Structured Data Manager - Discovery

Software Version 24.3.100

SDM - Discovery Guide

opentext™

Legal notices

Copyright 2022-2024 Open Text

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for updated documentation, visit https://www.microfocus.com/support-and-services/documentation/.

Support

Visit the MySupport portal to access contact information and details about the products, services, and support that OpenText offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- View information about all services that Support offers
- Submit and track service requests
- Contact customer support
- Search for knowledge documents of interest
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in.

Contents

Introduction

Personal data (personally identifiable information (PII) or sensitive personal information (SPI)) is information that can be used on its own or with other information to identify, contact or locate a single person, or to identify an individual in context. Any information that can be used to distinguish one person from another or could be used for de-anonymizing anonymous data is also considered personal data.

About Discovery

The SDM Discovery feature:

- Enables you to define discovery projects that analyze, declare, and manage sensitive data. The discoveries based on business needs can be grouped together into projects.

- Enables the user to document details of where personal data resides within corporate databases and other structured repositories.

- Discovery also assists in the planning and design of processes to manage this data to help comply with data retention requirements and data privacy rules.

- It uses the database connection (OLTP_DB location) from the SDM environments to retrieve the list of source tables and to scan the content of those tables.

Discovery Overview

Before you begin to use Discovery, you must ensure that SDM , SDM Postgresql repository are installed, and a source environment is available.

1. See, Concepts to understand more about Discovery and its features.

2. See, Installation on how to install Discovery and Discovery services.

3. See, Data sources on how to add data source and create environments.

4. See, Discovery workbench to start using Discovery from user interface.

5. See, Grammar workbench to create customized grammars.

6. See, Command Line Interface to understand how to use Discovery via command line interface.

7. See, Roles and Privileges to understand the roles and privileges for a user to use Discovery.

Concepts

SDM Discovery supports the discovery of customer's personal and sensitive data found in databases such as Oracle, SQL Server, DB2, PostgreSQL, or any accessible through JDBC.

This document explores the main Discovery concepts.

- Projects

- Schemas

- Tables

- Columns

- Grammars

- Risk score

- Review

Projects

A project is an organizational folder that will collect a list of schemas and tables to scan and review. A project represents a business area, a module of a complex ERP application or a simple application. A project will have a risk score and is the unit of work for Discovery. Therefore, it is important to understand that a project must contain a limited number of tables (no more than 1000) that represent the business area.

For example, imagine you want to scan Oracle eBusiness Suite. OEBS covers 9 business areas.
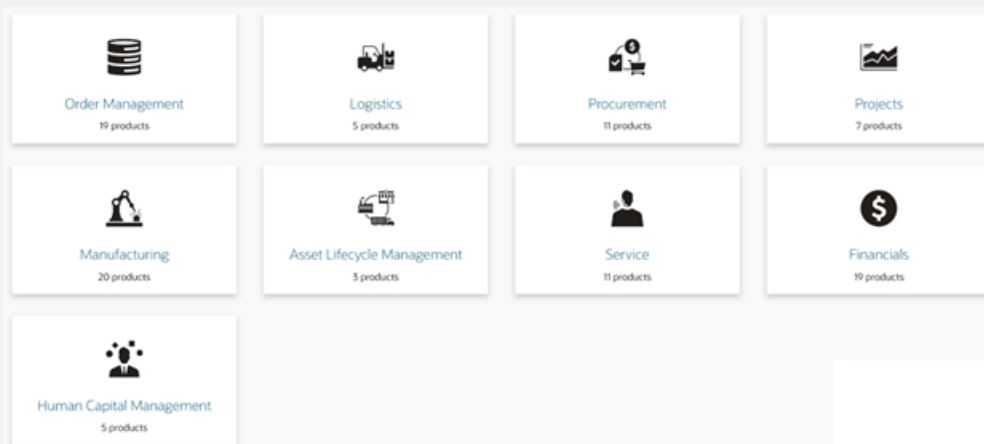
1. Order Management (19 modules)

2. Logistics (5 modules)

      a.   Inventory

      b.   Supply Chain

      c.   Yard Management

      d.   Landed Cost Management

      e.   Warehouse Management

3. Procurement (11 modules)

4. Projects (7 modules)

5. Manufacturing (20 modules)

6. Asset Lifecycle Management (3 modules)

      a.   Assets

      b.   Enterprise Assets

      c.   Complex Maintenance Repair and Overhaul

7. Service (11 modules)

8. Financials (19 modules)

      a.   General Ledger

      b.   Payables

      c.   Receivables

      d.   Payments

      e.   Treasury

      f.   Property Management

      g.   Cash Management

      h.   Lease and Finance Management

      i.   Loans

      j.   Credit Management

      k.   ETC.

9. Human Capital Management (5 modules)

      a.   Talent Management

      b.   Workforce Management

      c.   ETC

Each business area covers a given number of modules or products. A project must represent a module in a business area. When you see a risk score you would know it is for Supply Chain or Talent management or other well defined domain area.

Another example in the telecom industry would have:

1. Resource scheduling

2. Service parts planning

3. Finance and regulatory

4. Sales and Distribution

5. Quality management

6. CRM

Each one of these will be subdivided modules.

NOTE:
- A project must collect ONLY the tables that play in a particular business area represented by it.

- If you have more than 1000 tables in a project, it probably means that you have not divided enough your business areas and you are mixing tables of different usages together.

Exporting and Importing Projects

The Discovery module allows the export of projects so that they can be transferred to another instance of SDM with the discovery module enabled. You must also ensure that the version of the discovery module is same between the origin and destination.

NOTE: The SDM repository user requires the SUPERUSER role in Postgres to enable the export or import option.

Remote Projects

Remote projects are the projects that are collected by different instances of SDM, they are exported from that instance and imported to another instance. This allows reporting to be done remotely from the scanning.

Some features are disabled for remote projects such as adding tables, performing, or scheduling scans. But you can still review the collected meta data and the review columns.

Schemas

A database schema represents a logical grouping of a relational database objects. A database schema collects tables and views and other objects. Schemas must be registered to a project to collect their tables.

Tables

A table is a collection of related data held in a table format within a database. It consists of columns and rows. Tables must be registered to a project to be part of it. A table is a unit of Discovery

scanning.

Columns

In a relational database, a column is a set of data values of a particular simple type, one value for each row of the database. A column may contain text values, numbers, or other more complex types.

Some relational database systems allow columns to contain more complex data types; whole documents, images or video clips are examples.

A column is a unit of Discovery review: a personal or sensitive type of data can only be found in a column.

> **NOTE:** Sometimes a data element is also called as a field.

Grammars

Grammars are what enables discovering personal and sensitive data.

The core components of Grammars are:

- Grammar classes
- Grammar rules
- Grammar types
- Grammar sets

Grammar classes

Grammar classes are categories of Grammar types. For example, they can group all financial types, or all contact types. Grammar classes have a weight. This weight will be used to calculate the risk score.

Grammar types and rules are collected in sets. Discovery module is installed with the Global Set that includes all types and rules. This set is called PII-def and will be the parent for any set you would like to create

Grammar sets

Grammar sets are collection of types and rules that can be assigned to a particular project.

For example, if you want to only collect FRANCE and WORLWIDE types and rules, you can create a set only for fr, and enable only ww and fr rules.

Grammar types

Grammar types are any type of data you need to discover. Examples of grammar types are "Credit Card", "First Name", "Last Name", "Social Security Number", etc.

- Grammar types have a risk level. This risk level will be used to calculate the risk score.

- A grammar type can be enabled or disabled.

- A grammar type will have 1 or multiple rules

Grammar Types	Description
List	Dictionary of words Useful for first names, last names, etc.
RegEx	- Regular Expression Useful for Credit Card, Bank Account, etc. - To a regular expression you can add a groovy method to reduce the number of false positives or do further precise scanning. - For example, you can add `isValidVisaCard` to make sure a credit card is a valid one. - This method needs to return true or false
Method	- When a regular expression or a list cannot be used you can create your own algorithm in a groovy method and supply it. - This method needs to return true or false
Range-number	Scan for numbers between 2 values
Range-date	Scan for date age (converted to years) between 2 values. Useful to find birth days
Range-string	Scan for number of words in a text between 2 values

Risk Level will be used in the formula that calculates the risk score.

Grammar rules

Grammar rules belong to a particular type and are rule that defines how to discover. Examples of grammar rules for type "Credit Card": "Visa", "American Express", "Master Card" etc.

- A grammar rule can be enabled or disabled.

- Grammar rules are for a given grammar type.

List	Datafile	Dictionary as file
	Query	Dictionary as SQL query
	Datamethod	Dictionary as groovy method returning a list of strings
RegEx	Pattern or Pattern+Method	Regular expression, plus optional groovy method returning a Boolean

Method	Method	groovy method returning a Boolean
Range-number	Range-number	Number between 2 values
Range-date	Range-date	Years between 2 values
Range-string	Range-string	Number of words between 2 values;

Risk score

A risk score is calculated at the column level and then aggregate to the table and to the project. Discovery will calculate

- Finding risk score: risk score for a particular grammar-type found on a given column
- Column risk score (sum of all found grammar-types risk scores)
- Table risk score (sum of all contained column risk scores)
- Project risk score (sum of all contained table risk scores)

```
Finding Risk Score = log10(samplehits+1)*typerisklevel*classweight
```

```
Column Risk Score = ∑ Finding Risk Score
```

```
Project Risk Score = ∑ Table Risk Score
```

```
Table Risk Score = ∑ Column Risk Score
```

> **NOTE:** For tables that have zero hit (e.g., table has no rows) and the name of the column matches an alias, the risk score is equal to 1.

Risk index

A risk index is a calculated value between 0 and 100 for a given column relative to the highest column risk score in each project.

In each project, the highest risk score column will have a risk index of 100.

For example,

Risk Score of columns, in Project P	Risk Index of the column
Highest risk score column H = 1000	100%
Risk score for column A = 500	50%
Risk score for column A = 200	20%

Review

First scan for a table:

Scan Status	Means	Action	Review Status
Suspect	Personal or Sensitive type(s) found	Confirm False Positive Over-ride	Sensitive Safe Sensitive
Review	NO Personal or Sensitive type found	Confirm Over-ride	Safe Sensitive

Subsequent scan for a table:

Scan Status	Means	Action	Review Status
Sensitive	SAME Personal or Sensitive type(s) found	Nothing to do	Sensitive
Safe	SAME Personal or Sensitive type(s) found	Nothing to do	Safe
Suspect	Personal or Sensitive type(s) found: Discrepancy between previous findings and current	Confirm False Positive Over-ride	Sensitive Safe Sensitive
Review	No Personal or Sensitive type found: Discrepancy between previous findings and current	Confirm Over-ride	Safe Sensitive

Installation

Installation in discovery consists of:

- Discovery installation

- Discovery Service installation

Discovery installation

> **NOTE:**
> - The Discovery project supports only PostgreSQL repositories and requires the **obtdiscovery** schema in the SDM repository. The schema gets created automatically the first time the extension runs.
>
>   > **IMPORTANT:**
>   > ○ If you install Discovery module on non-PostgreSQL repository, the module behavior is unpredictable.
>   >
>   > ○ Prior to SDM 7.61, the Discovery module was made available by default, hence during upgrade to SDM 7.63 or above, you will see a tile to Install or Upgrade Discovery, you can ignore the same.
>
> - The Discovery project supports user privileges. With these user settings, some of the users might have access to only a few environments while others might have access to all environments. Based on the user privileges you can access the respective environments in Discovery. For more information on SDM Web Console users, see *Structured Data Manager Runtime Guide.*

To install the **Discovery Module 24.3.100**, either be a fresh install or an upgrade from an earlier version, follow the below steps:

1. Ensure that the following prerequisites are met:

   - **SDM 24.3.0 must be installed**

   - SDM PostgreSQL repository must be installed.

   - You have to uninstall the Discovery service while you are in the previous SDM version. See Discovery Service installation to re-install the service after upgrade.

2. Shutdown the Web Console if running.

3. Run the following commands in the command prompt window:

   - **On Windows**:

     ```
     ...> cd <Install_dir>\obt\bin
     ```

     ```
     DiscoveryInstaller.bat
     ```

- **On Linux**:

  ```
  ...> cd <Install_Dir>/obt/bin
  ```

  ```
  sh DiscoveryInstaller.sh (or) ./DiscoveryInstaller.sh
  ```

  > To force a new install run `DiscoveryInstaller.bat forceNewInstall`.

4. Launch Discovery in any of the ways below

   a. User Interface

      i. Start the Web Console

      ii. Click Discovery tile

      This opens the Discovery in UI mode. See Discovery workbench to continue using Discovery.

   b. Command Line Interface

      i. Run the below command from the specified path:

         - **On Windows**:

           ```
           > cd <Install_dir>\obt\bin
           ```

           ```
           Discovery.bat -a
           ```

         - **On Linux**:

           ```
           > cd <Install_dir>/obt/bin
           ```

           ```
           sh Discovery.sh (or) ./Discovery.sh
           ```

      ii. Enter the encryption created during the repository creation.

          > **NOTE:** The encryption key is required only once when you first launch Discovery.

          This opens the Discovery in CLI mode. For more information on how to use Discovery CLI, see Command Line Interface.

> **IMPORTANT:** If you have an environment other than DB2, Oracle, Postgres, SQL Server, or Sybase then you must manually copy the database specific JDBC driver jar files to `<SDM Install Dir>/obt/webconsole/apachetomcat/webapps/WebConsole/WEB-INF/lib`.
> To do so, perform the following steps:
> 1. Shutdown the Web Console.
>
> 2. Copy the JDBC driver jar file(s) to `<Install_dir>/obt/webconsole/apachetomcat/webapps/WebConsole/WEB-INF/lib`.
>
> 3. Start the Web Console.

Discovery Service installation

> **NOTE:** Ensure to run Discovery Service with administrator privileges.

To install Discovery Service, follow the below steps:

1. Open command prompt, and run the below commands

    On Windows:

    ```
    > cd <Install_dir>\obt\bin
    ```

    ```
    Discovery.bat -a
    ```

    On Linux:

    ```
    > cd <Install_dir>/obt/bin
    ```

    ```
    sh Discovery.sh (or) ./Discovery.sh
    ```

2. Enter the encryption key that you had provided during the repository creation.

3. In the command prompt, type q and press **Enter**.

4. Run the following command in command prompt to install and start the SDM Discovery Service.

    On Windows

    ```
    >cd <Install_dir>\obt\bin
    ```

    ```
    DiscoveryService.bat install srvid
    ```

    On Linux:

    ```
    >cd <Install_dir>/obt/bin
    ```

    ```
    sh DiscoveryService.sh start srvid (or) ./DiscoveryService.sh start srvid
    ```

    > **NOTE:**
    > - srvid must be between 3 and 9.
    >
    > - It is added to file dscServiceid.env file.
    >
    > - If srvid is not specified, then the default value is taken as 3 or from the srvid in dscServiceid.env file under <Install_dir>/obt/bin.
    >
    > - If SDM is integrated with Fusion platform in the Cloud,
    >
    >   - Edit the <OBTHOME>/config/fusionapp.properties file and set service.scan.id=srvid.
    >
    >   - It is used for scan requests from the Cloud.

- To **start, stop, or check status of the service**, run below commands

    On Windows:

```
> cd <Install_dir>\obt\bin

DiscoveryService.bat status srvid

DiscoveryService.bat start srvid

DiscoveryService.bat stop srvid
```

On Linux:

```
> cd <Install_dir>/obt/bin

sh DiscoveryService.sh status srvid (or) ./DiscoveryService.sh status srvid

sh DiscoveryService.sh start srvid (or) ./DiscoveryService.sh start srvid

sh DiscoveryService.sh stop srvid (or) ./DiscoveryService.sh stop srvid
```

- **To delete the service**, run the below command

  On Windows:

```
> cd <Install_dir>\obt\bin

DiscoveryService.bat delete srvid
```

Data sources
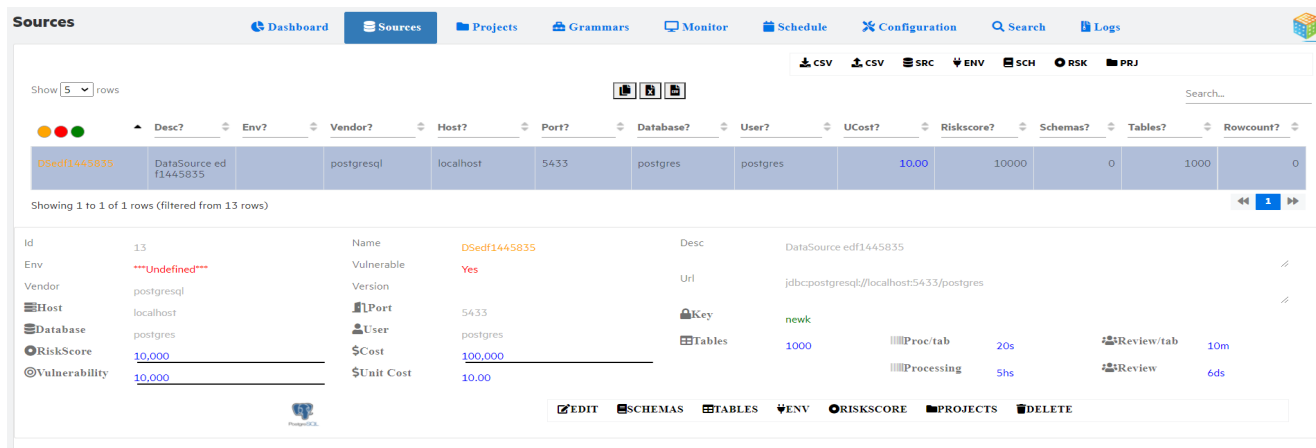
Introduction

The environments created from SDM are automatically populated as the datasources. Many datasources can be created at the same time by importing or exporting datasource details from / to a CSV file.

To add a new data source and create environment on it, follow the below steps:
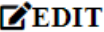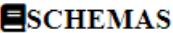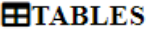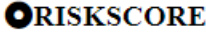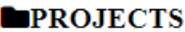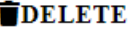
1. Click **SRC** to add a new source

2. Enter the details of datasource and a key name

3. Select the newly created datasource and click **Key** to update the datasource password

4. Click **ENV** to create environment on the selected datasource

   After successful creation of datasource and environment you can collect schemas, tables, get risk score and synchronize projects on the datasource.

All the operations on datasource can also be performed using CLI commands. For more information, see datasource Commands

| Button /Field	Description
🟠 🔴 🟢	Filter datasources based on the following: • All datasources • Datasources with no environment

|	
	• Datasources with environment
⬇ **CSV**	Import data sources from CSV file. The file name of the csv file and the location from where the data source is imported must be `datasources.csv` and `<OBTHOME>/obtdiscovery/csv directory` respectively.
⬆ **CSV**	Export data sources to CSV file The data sources are exported to location `<OBTHOME>/obtdiscovery/csv directory` with file name `datasources_export.csv`

▤ **SRC**	Add new data source
	
	Enter the following to add new data source
	
	• Name
	
	• Desc
	
	• JDBC provider
	
	• Host
	
	• Port
	
	• Database
	
	• User
	
	• Key
	
	    ◦ Enter the key name whose value indicates the source password.
	
	    ◦ If the key is already created, use the same key name else create a new key name, and update key value using 🔒**Key** after the datasource creation.

🔌 **ENV**	Create missing environments > **NOTE:** Make sure that the data source details and the keys are correct for the datasources. Check the **Submit job** check box to start creating the missing environment for the data sources
📖 **SCH**	Collect Schemas This collects all the schemes for the datasource, make sure all the datasource is available for it to work.
⊙ **RSK**	Refresh Riskscore

	This refreshes the risk score for each data source based on the projects at that source
PRJ	Sync Project This synchronizes all the projects and data sources in discovery
Key	Select the key option to edit a key value for the data source. **Key value** stores the password value of the data source. Click ✋✋ to add a new key to the bag.
EDIT	You can edit the description, key, and processing parameters for the selected source.
SCHEMAS	Collects the schema for the selected data source.
TABLES	Collect tables for the selected data source.
ENV	Create environment for the selected data source. **NOTE:** Make sure that the data source details and the keys are correct for the datasource.
RISKSCORE	Refresh the risk score for the selected data source.
PROJECTS	Synchronize the projects of the selected data source.
DELETE	You can delete a datasource which do not have any environment for it. **NOTE:** If you delete an environment from SDM, you must manually delete its corresponding datasource from Discovery.

Main Dashboard
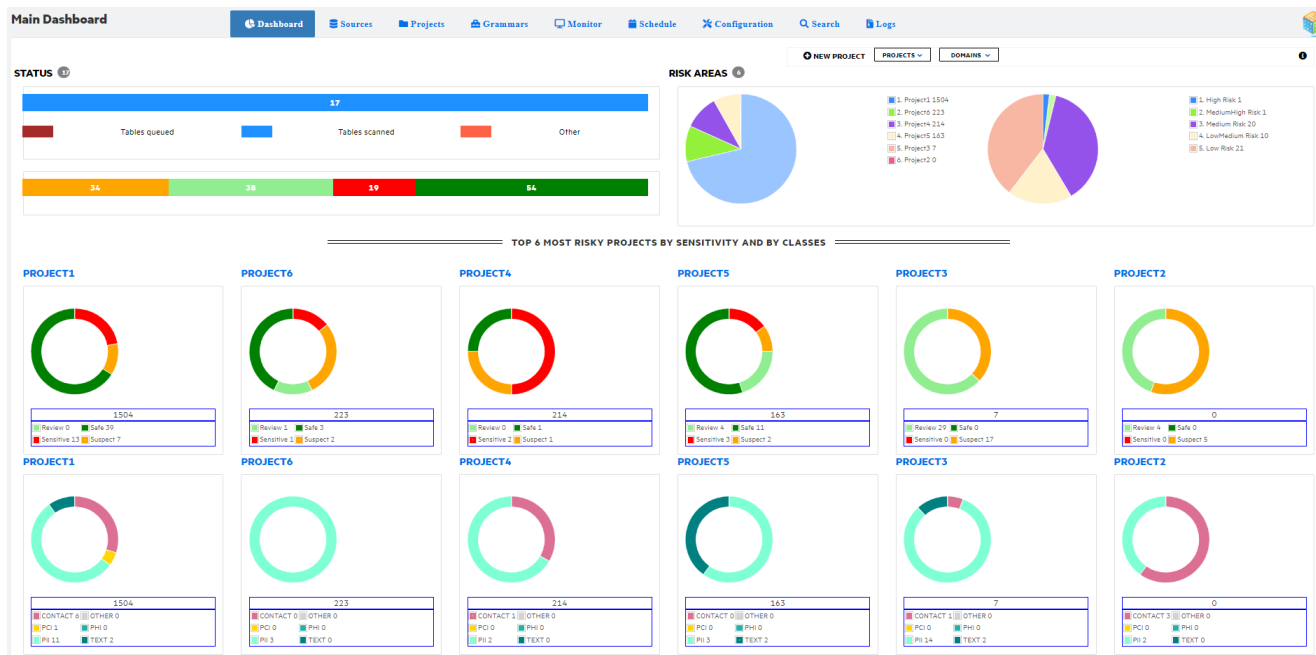
- **Status** - Displays the number of tables based on status as queued, scanned, other (frozen, erred, or currently scanning) and the number of columns based on the review type as sensitive, safe, suspect, and review. Click **Status** name to view the table review status and sensitive count field of all the projects not having a domain.

- **Risk areas** - displays the risk area for each project. Click **Risk area** name to view the table review status and sensitive count field of all the projects having a domain.

- Main dashboard displays only the top six most risky projects by sensitivity and by classes.

  - **Table review status** - displays table review status for each project with tables classified as sensitive, suspect, review and safe. Click any project name to navigate to its dashboard.

  - **Sensitive count field** - displays class of sensitive data for each project.

Button/ Field	Description
New Project	See Create a discovery project
Projects	List all the available projects. You can click on a project to go to its dashboard
Domains	List all the domains of the projects. Select a domain to view the table review status and sensitive count field of all the project of that domain.

ℹ	Explains about Discovery and the installed configuration details such as version number, last update, and language.

Discovery workbench

Discovery workbench explains in detail about:

- Projects
- Project schemas
- Project tables
- Analyze Registered Tables
- Column Dashboard
- Project Audit
- Project Monitor
- Project Report
- Reorganize

Source Environment

A source environment is a connection to a database instance. You must have created at least one source environment to create a Discovery Project

Credentials used to login to the source need to have

- **read catalog**: ability to interrogate database catalog metadata
- **select privileges**: ability to read rows from a given tables

Projects

To define a project, you must first Create a discovery project . Once a project is created, the Project Dashboard page for the new project opens. You can view the entire lists of projects created in Project List page.

Create a discovery project

Button/Field	Description
New Project	1. Navigate to the **Discovery** module in the Web Console. 2. Click **New Project.** The New Project dialog opens. 3. In the **Project Name** box, enter a project name.

> **NOTE:** A project name is required. The project name must be unique.

4. Enter an optional description for the project in the **Description** box.

5. Select an environment from the **Environment** list.

> **NOTE:** The word **remote** is a reserved word and cannot be used for the environment name.

6. Click **Save.**

> **NOTE:** To import a project created on a different SDM instance, see the section on Import Project

Project Dashboard

The Project dashboard page for the new project opens, displaying a summary of the project, project name, environment, creation date, status, database type, grammar set and description.
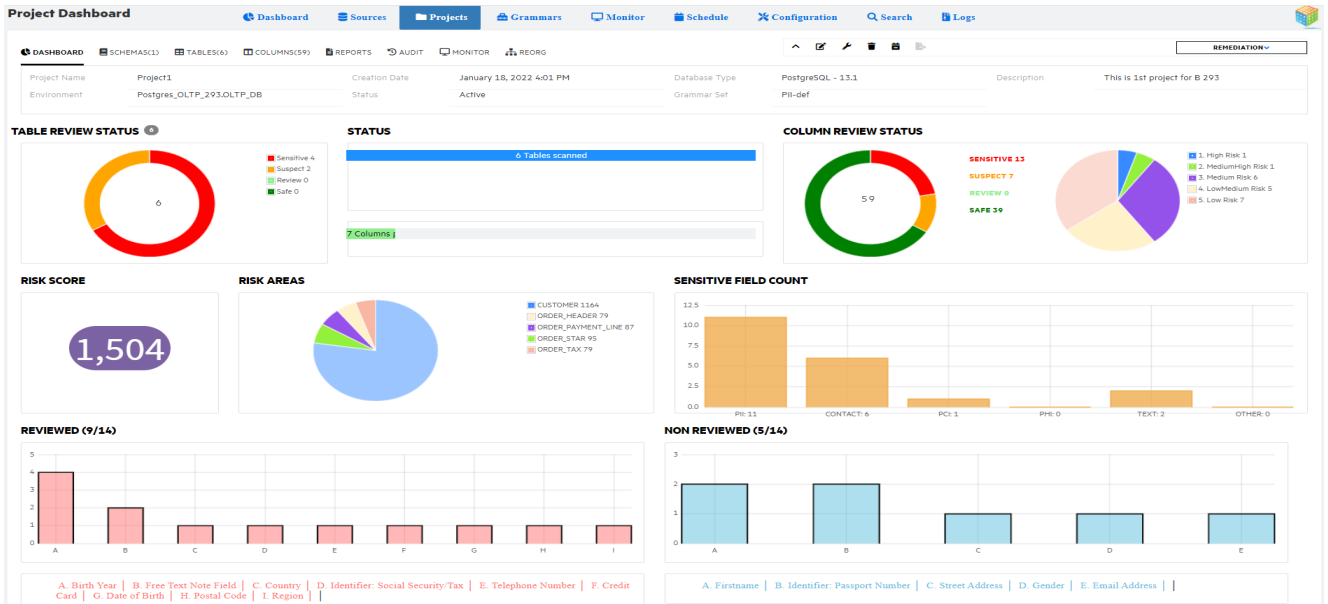

Button/Field	Description
Up a level	Displays the project list pages.
Edit Project	The edit option allows you to edit the following settings on the Discovery project. • **Project Name**. The name of the project. • **Project Description**. The description of the project. • **Grammar Set**. Enter the set of grammars used for scanning. Any alterations done on the grammar set forces a project reset and requires a new scan to be performed. • **Domain, Group, and Application**: Enter the name of domain, group, and application the project belongs to. • **Region**: Enter the country /region in which the application is used. • **Email**: Enter the email of the user/company.

Button/Field	Description

	• **Source Type**: Select the source type for Discovery.
	
	   ◦ Production
	
	   ◦ Test
	
	   ◦ QA
	
	   ◦ Development
	
	   ◦ Education
	
	• **Masking Use**: Enter the **Masking use 1 /Masking use 2** which is used for grammar masking strategies.
	
	• **Project Type**: Enter the type of project.
	
	• **Show on Dashboard**. Click the **Show on Dashboard** option to show the project or to hide the project from the main dashboard.
	
	• **Lock Project**. Click **Lock Project** to prevent any changes to the project. This includes disabling a scheduled scan, preventing deletion, and allowing export.

Project Scan Settings	You can set the scan parameters at the project level, which will override the global configuration of the scanner.
	
	• **Register DB Types**.
	
	   ◦ **Inherit**. Inherits the global values.
	
	   ◦ **TABLE**. Allows to register only table.
	
	   ◦ **VIEW**. Allows to register only view.
	
	   ◦ **TABLE, VIEW** Allow to register both the table and view.
	
	• **Project Service**. Enter the id of Discovery service to be used by the project during scan. See Discovery Service installation to install Discovery Service. The scans registered for schedule on Discovery service use the id specified here.
	
	• **Project Threads**. Enter the number of threads to be used by the project.
	
	• **Count SQL**. Row counts of the table or view registered.
	
	• **Sample SQL**. SELECT query that is used for the sample during scanning.
	
	NOTE: The default **Sample SQL** query does not work for scanning SQL Server, Sybase and PostgreSQL view and you must modify the query. See Scanning of views to modify the Sample SQL query.

Scan Blackouts	• You can set scan blackouts time for particular project.

Button/Field	Description
	• There will be no scanning done for this duration.
	• Specify day and time where no scanning should take place.
	• You can have scan blackout globally or based on the database.
	• You can click Set no constraints icons and choose blackout time from 6-18, 7-17, 8-18, 18-23, 19-23 and 20-23.
Export Project	A locked project can be exported to allow transport to another SDM instance. 1. Expand **Project Options** and select **Export**. Export only becomes available after you have locked your project indicated by a lock icon next to the project name (see Edit Project for details on locking projects). 2. Click the **Export** button on the popup dialog to start the download, and your project will be downloaded in a zip file. You must ensure not to edit or rename this file.
Import Project	On the main dashboard or projects tab, do the following: 1. Click the **New Project** button. 2. Click the **Import Project** switch. 3. Locate your exported project file using the **Choose file** button to open a browse dialog, click **Open** and then **Save**. During the import, if an environment exists that matches the project environment, then the project is attached to this environment. If the environment does not exist, then the project is imported as remote.
Delete Project	Select to delete a project.

Project dashboard details

After the table and column review is completed, you can see a detailed project dashboard displaying table review status, status, column review status, risk score, risk area and sensitive field count including a donut graph of the records by classification.

Tabs	Description
Table review status	Displays the number of tables that are in sensitive, suspect, review, and safe state. When the table is scanned for the first time, you will have only two types of status, namely, suspect and review. During scanning, the table can have the following types of status: • Sensitive • Suspect • Review • Safe While scanning a table, if the data in a column is found to be sensitive then the column is flagged as suspect. This means that the data in this column needs to be reviewed. A user or admin needs to look at the data that is flagged as review. Once reviewed, the data can then be moved to either sensitive or safe status.
Status	Displays the number of tables that are in queued, scanned, frozen state and number of columns that needs to be reviewed.
Column review status	This is similar to table review status but on a column level and has sensitive, suspect, review, and safe types of status.
Risk score	Displays the risk score for the project.
Risk areas	Displays the top ten tables that are flagged as risky tables.
Sensitive field count	Displays the field count based on the class of sensitive data.

Remediation	Remediation allows you can create a designer project, generate masking functions, composition functions, and SQL statements for a project.
	Navigate to project dashboard to access the following remediation options:
	• Click on any of the remediation option, displays the respective dialog box.
	◦ Designer project
	◦ Generate masking
	◦ Generate compositions
	◦ Generate shuffle SQL
	◦ Generate shuffles
	• **Confirm** and click **Generate**.
	Once the operation is complete, a path is displayed where the generated output is saved. You can further act on the data as per your requirements.

Cloud manged project

On a project dashboard page, if you see an alert as shown in the figure below, it is recommended not to make any changes to that project from the Discovery UI because it is a cloud-managed project.
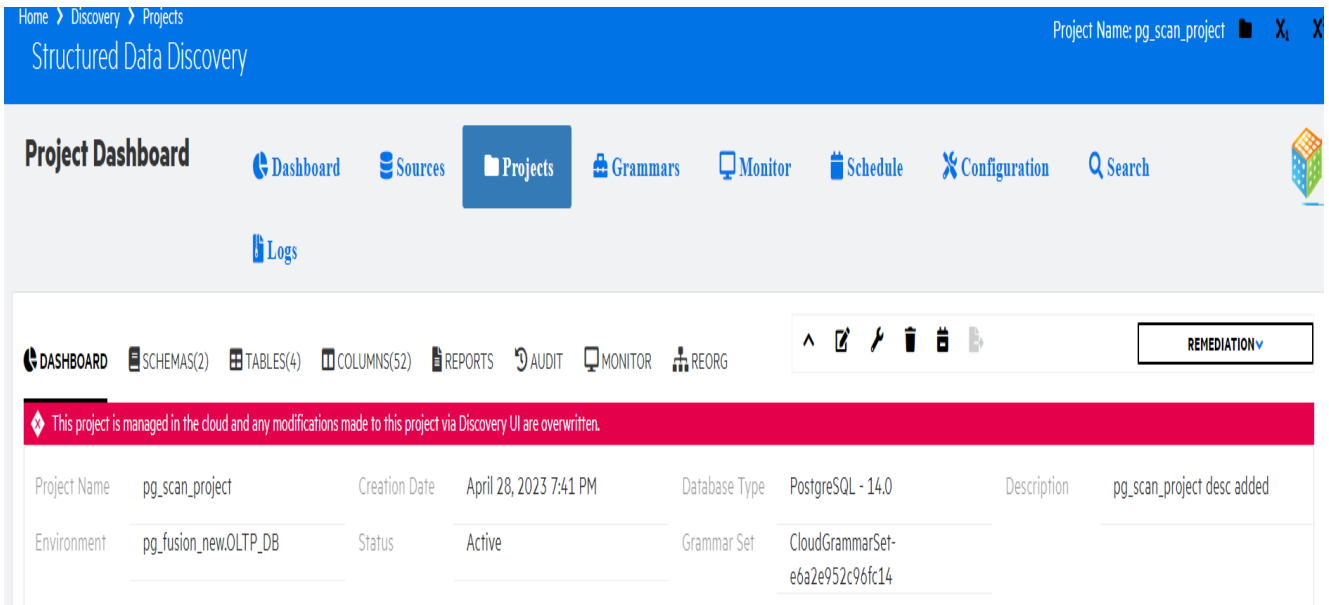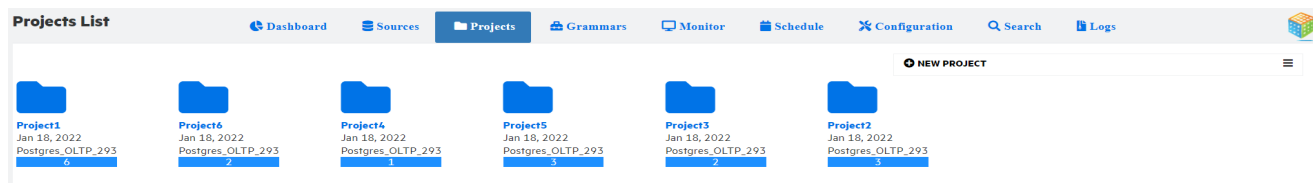
Project List

This page displays the projects lists view as icons or tables. To configure the default view for project list, see Project View
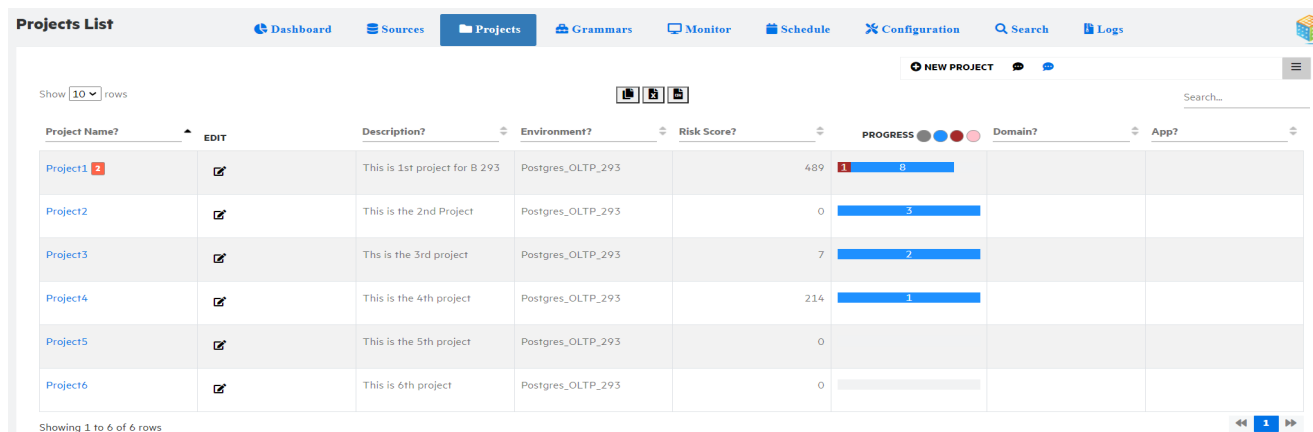
- **Icons**

  This view lists the projects as icons with project name, the date on which the project was created and the environment.

  - Click on project to display the respective project dashboard.
  - Click **New Project** to create a new project.
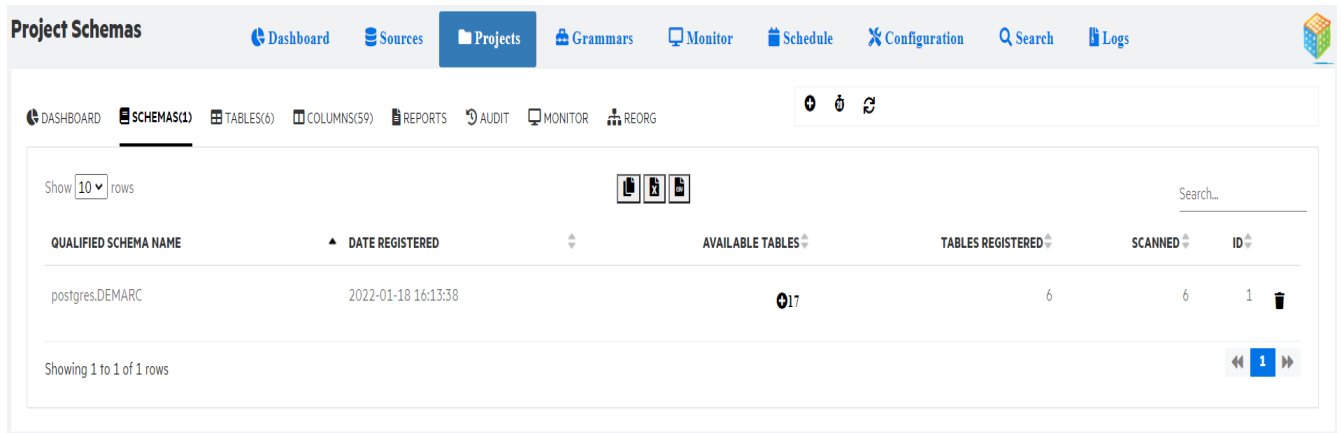
- **Table**

  This view list the projects in table with project name, edit, description, environment, risk score, progress, domain etc.

  - Click on project name to display the respective project dashboard.
  - Click **New Project** to create a new project.
  - Click on **Show / Hide Details** to show and hide the columns in the table view of the project list.
  - Click on icon to edit the project details.
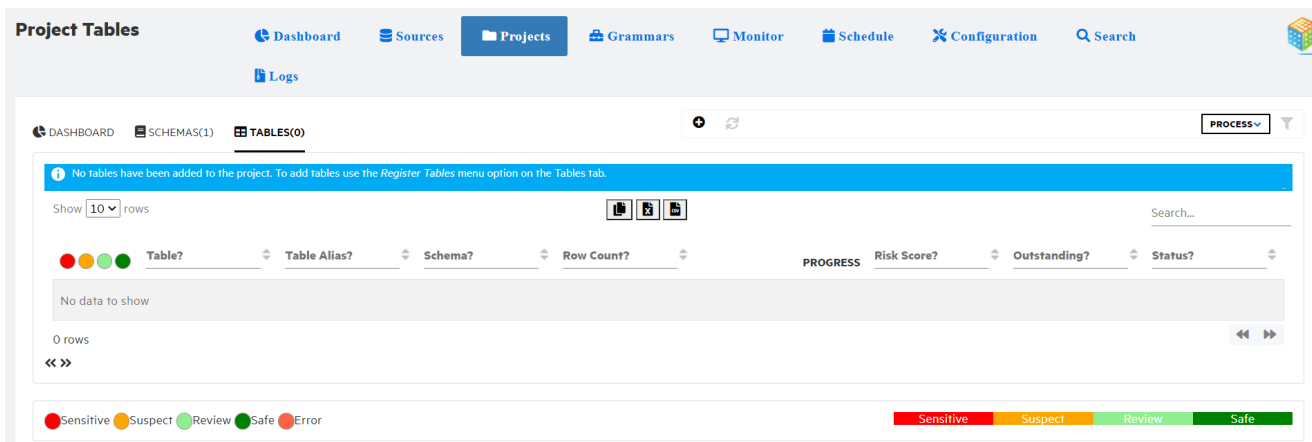
Project schemas

You must first register the schemas that contain the tables you want to scan.

To register schema, follow the below steps:

Button/Field	Description
Register Schemas	This option lists all the available database schema in the Register schema dialog 1. Navigate to the Project where you want to add schemas. 2. Click **Schemas** tab in the project dashboard page. 3. Click **Register Schemas** button. 4. In the list of Schemas, search and select the Schemas that you want to add. You can select multiple schemas to add at the same time. 5. Click **Save** button. 6. The schemas tab updates with the schemas you have registered.
Count Tables	Counts the number of tables for all the registered schemas.
Refresh	Refreshes the page or lets you select an interval at which the page refreshes.
Delete schema	Deletes the selected schema.

Project tables

To define the tables, you must add tables to be scanned available in schema following the below steps:

Button/Field	Description
Register Tables	1. Click **Tables** tab in the project where you want to add tables. 2. Click **Register Tables** button. 3. In the list of Tables, search and select the tables from the list of **Qualified Table Name** that you want to add. 4. You can select multiple tables to add at the same time or use the **Add All Available Tables** to add all tables that is not added to the project. **NOTE:** • By default, the maximum number of tables that can be registered at a time is 5000. • You can change the default value in the **Configuration > UI > Register Tables Limit**. 5. Click **Save** button. The tables tab updates with the tables you have registered.
Refresh	Refreshes the page or lets you select an interval at which the page refreshes.
Advanced Filters	Click the icon to filter the information based on schema name, table scanned already, tables that need review, sensitive columns, and other filter fields.

Scanning of views

You can modify the default **Sample SQL** only via CLI.

To modify the Sample SQL query, perform the following

1. Open the CLI mode

   On Windows

   Navigate to `<Install_dir>\obt\bin` and type `Discovery.bat -a`

   On Linux

   Navigate to `<Install_dir>/obt/bin` and type `./Discovery.sh`

2. Set the query for your database.

   Examples:

Database	Query
PostgreSQL	`set param sample:postgresql=SELECT &C& FROM &T& ORDER BY RANDOM() LIMIT &N&`
SQL Server	`set param sample:microsoftsqlserver=SELECT TOP &N& &C& FROM &T& ORDER BY NEWID()`
Sybase	`set param sample:adaptiveserverenterprise=SELECT TOP &N& &C& FROM &T& ORDER BY NEWID()`

3. Trigger scan.

> **NOTE:**
>
> - You can modify **Sample SQL** both at global level and project level
>
> - Use command `set param sample:postgresql`to modify **Sample SQL** at global level. This global setting is applied to all projects created on PostgresSQL environments.
>
> - Use command `set project samplesql <samplesql>` modify Sample SQL at project level.
>
> - **Sample SQL** for different databases are stored in Discovery. You can find the these sample queries in **Configuration** > **Sampling**.

Analyze Registered Tables

After defining the scope of the project, you must then analyze the scope of the tables. When tables are added to a project, they are inserted into a processing queue. These can either be manually processed one by one, or processed in one pass, or a schedule process can perform the processing.

Button/Field	Description
Process selection	Depending on the status of the table, you can further process the tables with the following operations: • **Reset** - resets the table to a state where it can be rescanned. • **Deepreset** - resets the table and removes all the history. • **Scan** - scans the table • **Freeze** - tables are flagged as frozen and cannot be scanned again. • **Delete** - delete the table > **NOTE:** You can delete tables only if you are an administrator. • **Tags** - tags the tables. This tag can later be used as keywords while searching. • **Comment** - add comments to the table
Processing rules	Displays the operations you can perform based on the table status. For example, if a table is in queued status, the operations you can perform are scan, delete and tag. • Click **Process** drop-down • Select **Help**. The Processing rules dialog contains the following tabs: **Rules**

Button/Field	Description

	The rules tab lists the operations that you can perform on the table depending on its status
	
	Statuses
	
	This tab explains different statuses a table can be in.
	
	A table can be in the following status:
	
	• **Queued** - a table can be in a queue
	
	• **Scanning** - a temporary state while a table is getting scanned
	
	• **Scanned** - table is scanned completely. Once the table is scanned, the table can be in the following status:
	
	   ○ **Review** - table needs to be reviewed to further process it as sensitive or safe.
	
	   ○ **Reset** - table can be reset which goes back to the queued state again.
	
	   ○ **Freeze** - table is said to be frozen and you cannot act on it further. To unfreeze the table, you must reset it to move it to its previous state, that is, scanned. If you need to scan the table again then you need to reset, it further.
	
	     NOTE: You can only freeze a table that is scanned.
	
	• **Error** - if a table is in error status, the details of the error will be available in the logs. Once you fix the error, reset the table to be in queued status and proceed further for scanning.
	
	Risk score
	
	This tab displays the formula for calculating risk score.

Scan Now	To **Scan** a single table 1. Click the Project **Tables** tab. 2. Check the table you want to SCAN. 3. Click **PROCESS > Now**. 4. To confirm the action, click **Scan**. You are redirected to the scan status page where you can follow the progress of the scan with the progress bars. To **Scan All** tables: 1. Click the Project **Tables** tab. 2. Click **SELECT > All On**

Button/Field	Description
	3. Click **PROCESS > Now**. 4. To confirm the action, click **Scan**. Or Click **Scan all tables** » icon You are redirected to the scan status page where you can follow the progress of the scan with the progress bars.
Scanner Status	Displays Project Monitor page showing the progress of table scan.
Scan Later	• To schedule a scan: 1. Select a table to be scanned. 2. Check the tables you want to SCAN or click **SELECT > All On**. 3. Click **PROCESS > Later**. 4. Select the **Reoccurrence > Once, Daily, Monthly, or Weekly** from the drop-down. 5. Click the calendar icon and select the date and click the clock icon and select the time to **Run on**. • To scan using Discovery Service: Enable the **Discovery Service** check box. The scheduled jobs are displayed on Schedule page.
Add Schedule Task	• To scan all the tables at the same time: Click **Add Schedule Task** icon on the **Tables** tab. • To scan using Discovery Service: Enable the **Discovery Service** check box. The scheduled jobs are displayed on Schedule page.
Mass Review	Click **Mass review** drop-down to review table (s) by singles or by risk index. • To review singles, 1. Select the table (s), 2. Click **Mass review** and select **Singles**. 3. Click **Review** in the dialog displayed. • To review by risk index, select the column(s), 1. Click **Mass review** and select **By RiX**.

Button/Field	Description
	2. Select options **Greater than Risk Index** or **Less than** in the dialog. You can also enter the risk index for Less than option. By default, it is 50.
Reset	**Reset a single table for rescanning** or **Reset all tables for re-scanning**: If you want to remove scan details from a table or all tables, run the scan again. 1. Click the Project **Tables** tab. 2. Click on **Reset**.
Deep reset	You can deep-reset a table to make the next scan the first one. If you significantly change the definitions of your grammars, you do need to deep reset all tables previously scanned.
Show /Hide details	Click **Show details** or **Hide details** to view the hidden details or hide details about the table.
Advanced Filters	Click the icon to filter the information based on hit ratio, null ration, tables that need review and sensitive columns.

Reviewing tables

The tables or columns can be reviewed through:

- Project level review

  - Project Tables > Project tables

  - Column Dashboard > Column Dashboard

- Table level review

  - Table Details > Project tables

  - Table Details > SELECT > All On > Project tables

- Column level review

  - Table details > Select the column > Project tables

  - Column details page > Project tables

Table Detail Page

The Table Detail tab displays a detailed report on the tables of the project.

Button/Field	Description

Process	After the tables are scanned and the data is ready for review, select the columns of interest and click **PROCESS** to perform one of the following review actions:
	
	 • To review column(s), perform the following:
	
	   1. Click **Process** and select one of the following from the drop-down:
	
	      ◦ Confirm
	      ◦ Override
	      ◦ False positive
	      ◦ Ignore
	      ◦ Comment
	
	   2. Select **Conclusion** and **Override Type** from the drop-down. These two are the mandatory fields indicated by a star.
	
	   3. Enter details in **Note**, **Origin**, **Shared**, and **Comments** as required.
	
	   4. Click **Save**.
	
	 • To view the column details,
	
	   Select **Drill Down** option from the **Process** drop-down.

Mass Review	To review column(s) by singles or by risk index.

Button/Field	Description

	- To review singles,
	
	   1. Select the column(s),
	
	   2. Click **Mass review** and select **Singles**.
	
	   3. Click **Review** in the dialog displayed.
	
	- To review by risk index, select the column(s),
	
	   1. Click **Mass review** and select **By RiX**.
	
	   2. Select options **Greater than Risk Index** or **Less than** in the dialog.
	
	   You can also enter the risk index for Less than option. By default, it is 50.

Actions	To perform any action on the selected table - Click **Reset**, **DeepReset** or **Freeze** in the confirmation dialog.
Table Audit	Displays Table Audit page.
Table sample settings	To define scan parameter settings 1. Click **Table sample settings**. 2. Provide **Sample Size**, **Sample Percent** and **Sample Max** for the table. 3. Click **Save.**

Table details	To edit table details, perform the following steps:
	
	1. Click **Table details** to display Table Details dialog.
	
	2. Enter name for Alias.
	
	3. Enter description, comments, and tags in the respective fields.
	
	4. Enter where clause for WHERE.
	
	  For example, CITY = 'BANGALORE'.
	
	5. Enter conditions for UNIONALL.
	
	  For example, cond1&&cond2&&cond3.
	
	> **NOTE:** You can either use where clause or UNIONALL conditions and not both.
	
	6. Click **Save**.

Scan log	Displays the scan history of the current table. - To download scan logs

Button/Field	Description
	Click **Scan log** and **Download**.
Show /Hide details	Click **Show details** or **Hide details** to view the hidden details or hide details about the table.
Navigation	Click **Previous Table** or **Next Table** to view its table details
Advanced Filters	Click the icon to filter the information based on hit ratio, null ration, tables that need review and sensitive columns.

Column detail page

The Column Detail tab displays a detailed report on the columns of the project. To view the details of an individual column:

1. On the **Project** Tables tab, click the name of the table.

2. Click the name of the column.

Button/Field	Description
Process	In some cases, bulk review of columns is required: 1. Navigate to the project **Columns** tab. 2. Click on the numbers in suspect and sensitive columns displays list of columns that needs review. 3. Check the **Column** you want to review or click **SELECT > All On**. 4. Click **Process** and select one of the following from the drop-down:

| Button/Field	Description
	• Confirm • Override • False positive • Ignore • Comment 5. Select **Conclusion** and **Override Type** from the drop-down. These two are the mandatory fields indicated by a star. 6. Enter details in **Note**, **Origin**, **Shared**, and **Comments** as required. 7. Click **Save**.
Column Dashboard	Displays Column Dashboard page
Column List	Display overview about the selected column
Table Details	Returns to the **Table Detail** page.
Navigation	Click **Previous Column** or **Next Column** to view column details.
View Data	If the system is configured to allow data viewing and if you are granted access to the View Data role, then you can view the samples of data fetched from the source database to assist reviewing the column.

Table Audit

Table audit page displays table history or details of the scans done on that table. It displays the table name, schema, table registered date, status of the table, count of rows in the table, scanned date, data sample, column count and log file.
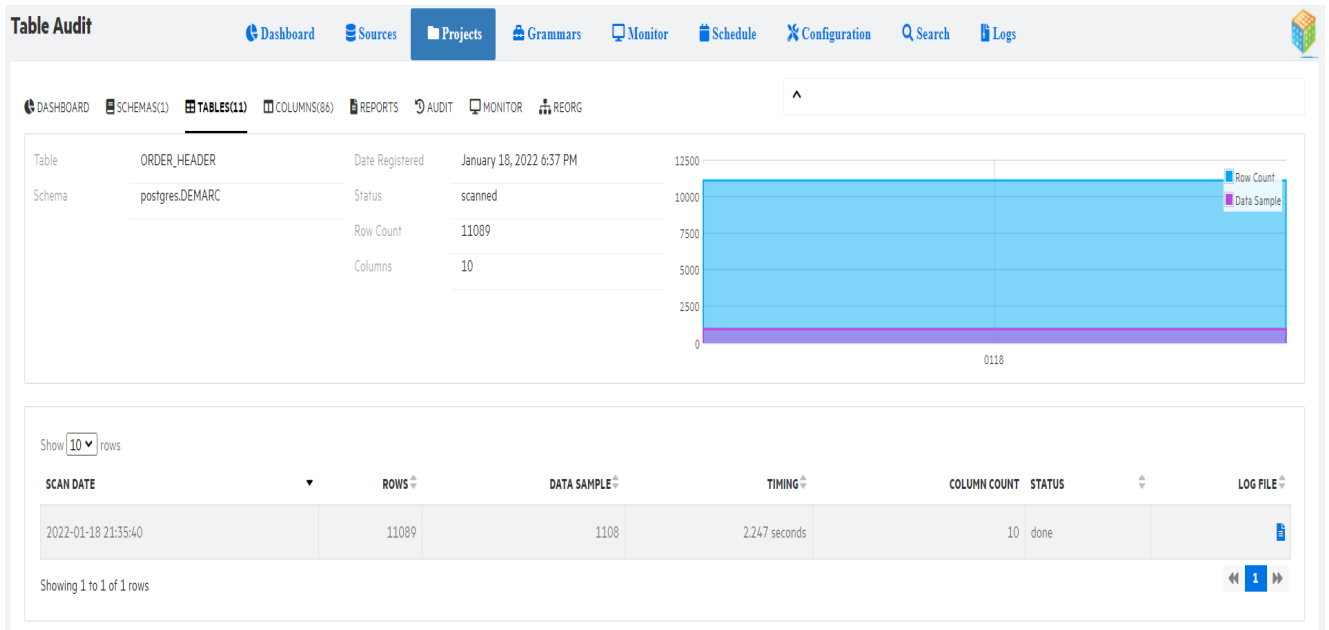
You can download log file to view the scan details of the table.

Column Dashboard

This page displays the statistics of the columns in your project, such as, column review status, project risk score, suspect and sensitive columns and scan results that includes reviewed and non-reviewed columns
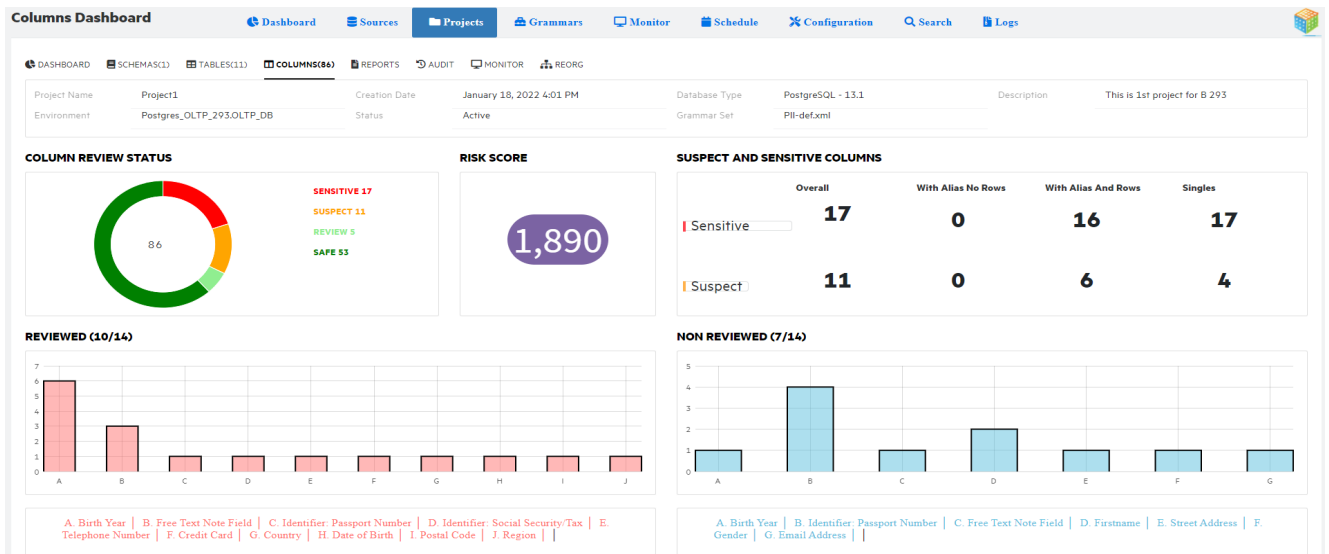
You can filter to view data from column dashboard. Selected column review page displays based on the filter applied.

- Click on the values in the chart, to filter the columns based on the status likes sensitive or safe

- Click on the numbers in suspect and sensitive columns to filter based on With Alias No Rows /With Alias And Rows/Singles

- Click on the bar graph / grammar type to filter the column based on the grammar type of reviewed and non-reviewed column

Selected column review

The selected column detail page displays a detailed report on the columns with same status.

Button/Field	Description

Process	**Bulk Review of Columns:** In some cases, bulk review of columns is required:
	
	1. Navigate to the project **Columns** tab.
	
	2. Click on the numbers in suspect and sensitive columns displays list of columns that needs review.
	
	3. Check the **Column** you want to review or click **SELECT > All On**.
	
	4. Click **Process** and select one of the following from the drop-down:
	
	  • Confirm
	  • Override
	  • False positive
	  • Ignore
	  • Comment

Button/Field	Description
	5. Select **Conclusion** and **Override Type** from the drop-down. These two are the mandatory fields indicated by a star. 6. Enter details in **Note**, **Origin**, **Shared**, and **Comments** as required. 7. Click **Save**.
Column Dashboard	Displays Column Dashboard page
Show / Hide details	Click **Show details** or **Hide details** to view the hidden details or hide details about the table.
Advanced Filters	Click the icon to filter the information based on hit ratio, null ration, size, tables that need review and sensitive columns.

Project Report

The Report page displays a detailed report on the status of the project.

Button/Field	Description
Non Sensitive Non Reviewed ◀	Lists all the tables which are non-sensitive and non-reviewed
Non Sensitive Reviewed ▶	Lists all the tables which are reviewed and non-sensitive

Button/Field	Description
Sensitive Non Reviewed ▲	Lists all the tables which are sensitive and non-reviewed
Sensitive Reviewed ▼	Lists all the tables which sensitive and reviewed.
Download HTML report 📄	Download the report in HTML format.
Download PDF report 📄	Download the report in PDF format.

Project Audit

The project audit page displays the audit logs on table scans and reviews. It contains date, log, user, table, and column. You can copy the audit log details or save the audit log as excel or .csv file.

Project Monitor

During an on-demand scan of all tables, the monitor page displays the progress of the scanner against the current table and overall progress of the scan.

It displays details of ongoing scans and the last scan completed for each table of a particular project. If an ongoing scan is taking long time or not progressing as expected, then you can stop the scan. When you stop a scan, the table is reset.

You can copy, download as excel or .csv the project scan status.

Button/Field	Description
All	Lists all the tables which are scanned, aborted, and erred
Done	Lists all the tables which are scanned successfully.
Abort	Lists all the tables which aborted during the scan
Error	Lists all the tables which erred during scan
Scanning	Lists the tables for which scanning is in progress

Reorganize

Reorganize page allows you to move table(s) from the current project to a target project. The target project must have same environment.
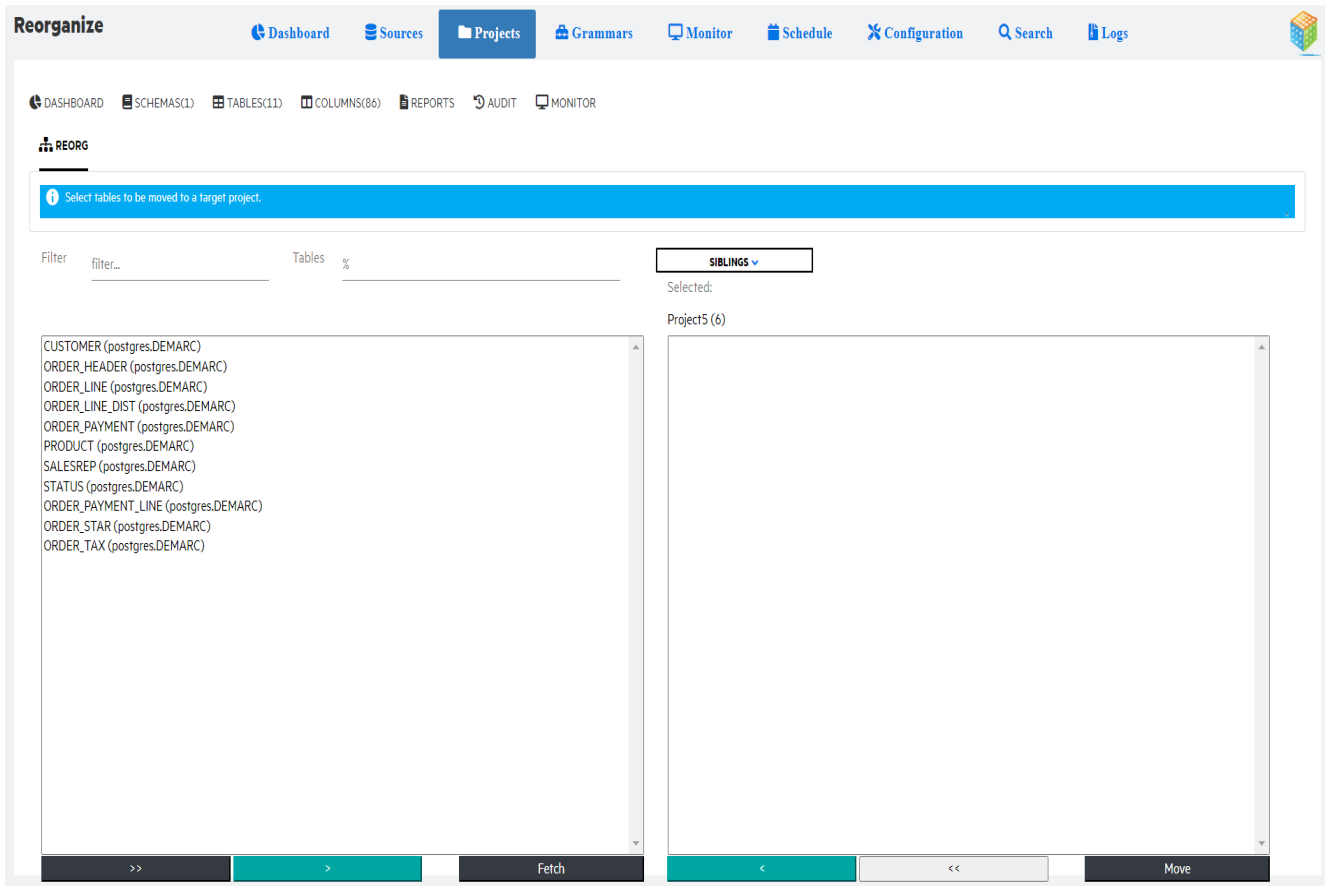
To move tables to a target project, perform the following steps:

1. Select the target project from the drop down.

2. Select the table(s) and use the arrows to move it under target project panel.

3. Click **Move**.

Grammar workbench

Grammars are defined through the UI and published in the form of an XML file. It is the XML file that will be used by a project when scanning any project table. You can assign a grammar set to a project. By default, PII-def is the grammar set that will be used.

The following are the core components of grammar:

- **Classes** - is a classification of grammar types. For example, personal, health, financial, and contact information.

- **Sets** - collection of types. You can assign a set to a project based on the country or locale. In the project, you can link a particular set with the project.

  Once you define sets, you can tune the rules or types as per your requirement.

- **Types** - helps you define sensitive information, like, social security number, credit card number, family name, passport name and so on.

  - **Rules** - every type is associated with one or more rules.

Grammar workbench describes about:

- Grammar dashboard
- Grammar definition files
- Grammar classes
- Grammar set
- Grammar types
- Grammar masking strategies

Grammar dashboard

The grammar dashboard lists the grammars, classes, and top locales.
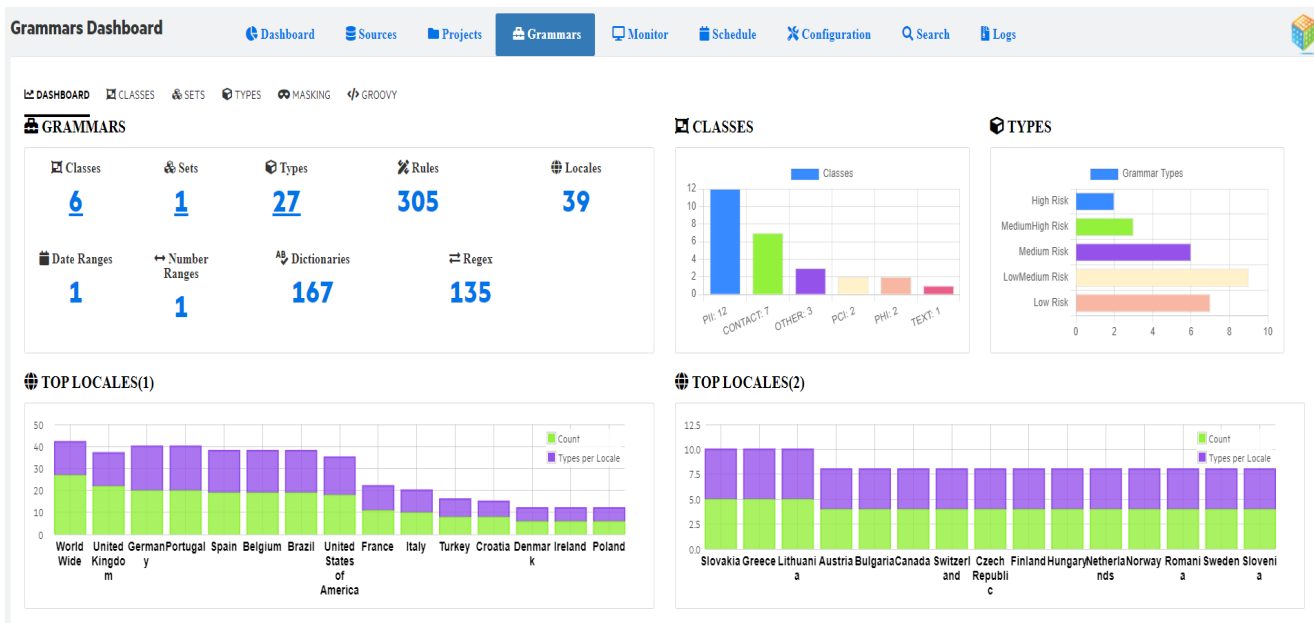
Grammar dashboard details

The following are the details of the grammar dashboard:

- **Grammars** - displays number of classes, sets, types, rules, locales, date ranges, number ranges, methods, dictionaries, and regex.
- **Classes** - displays the types per class
- **Top locales** - displays the count and types per locale

Grammar definition files

The grammar definition files that are effective can be found in the installation folder of SDM:

`<OBTHOME>\config\obtdiscovery`

A copy of the grammar definition files can be found in the following folder, this is just a backup and can be used to restore in case it's necessary:

`<OBTHOME>\extensions\webconsole\menus\main\Discovery\resources`

The following files are important for the grammar definition:

File name	Description
PII-def.xml	This file contains all grammar rules / definitions
\dictionaries*.txt	The files in the dictionaries folder are the grammar definitions that are based on a list of values (cities / names / countries etc.)
\extensions\	This folder contains example groovy scripts that for example can be used to enable validation functions.

To make changes to the grammars that are used for discovery the PII-def.xml file can be updated. Additional dictionaries can be added to the dictionaries folder or existing dictionaries can be updated. Additional validation functions can be added to the 'validate.groovy' script located in the 'extensions' folder.

PII-def.xml layout

The PII-def.xml file is used to define all the grammars. The file has a specific layout and elements within that will be shortly explained in this chapter.

The XML file starts defining the different PII classes that are distinguished with a 'PIICLASSES' tag. The image below contains an example of a PIICLASS definition. The following elements can be distinguished in this definition:

Attribute	Description
PIICLASS name	this attribute contains the generic name of the class
PIICLASS class	this attribute contains the 'classification' of the PIICLASS. Any grammar that is indicated to be part of this classification will be linked with this PIICLASS
PIICLASS weight	the height of the risk that values in the source database matching grammars of this type brings. The higher the risk the higher the weight.
TRANSLATIONS	is used to list different translations of the name of the PIICLASS and is used when the UI is running under a different locale than English.
TRANSLATION language	indicates for which UI locale this translation should be used.

```xml
<PIICLASS name="Personal Information"          class="PII"       weight="10" >
  <TRANSLATIONS>
    <TRANSLATION name="Personal Information"       language="en" />
    <TRANSLATION name="Info Personnelles"          language="fr" />
    <TRANSLATION name="Informacion Personal"       language="es" />
    <TRANSLATION name="Informazione Personale"     language="it" />
    <TRANSLATION name="Informação Pessoal"         language="pt" />
    <TRANSLATION name="Persönliche Angaben"        language="de" />
    <TRANSLATION name="Persoonlijke informatie"    language="nl" />
    <TRANSLATION name="Kisisel bilgi"              language="tk" />
    <TRANSLATION name="Informações pessoais"       language="br" />
    <TRANSLATION name="Persoonlijke informatie"    language="be" />
  </TRANSLATIONS>
</PIICLASS>
```

The next element in the `PII-def.xml` is the PIITYPES definition. Within this element all the grammar definitions are done, each definition is contained within a PIITYPE tag. The image below contains an example of a PIITYPE definition. The following elements can be distinguished in this definition:

Elements	Definition
PIITYPE name	this attribute contains the generic name of the class
PIITYPE type	this attribute contains the type of this grammar. The different types will be explained later in this document.
PIITYPE class	this attribute links the PIITYPE to the PIICLASS which it belongs to.
PIITYPE threshold	this attribute indicates the percentage of positive matches the grammar must have against a column of the database to be considered relevant.
PIITYPE risklevel	this attribute indicates the risk associated with this specific PIITYPE
PIITYPE mask	this must contain a unique value; it is only a technical value used internally.
PIITYPE id	each PIITYPE must have its own unique ID
TRANSLATIONS	is used to list different translations of the name of the PIITYPE and is used when the UI is running under a different locale than English.
TRANSLATION language	indicates for which UI locale this translation should be used
ALIASES	a list of column names in the source database that commonly indicates the presence of values of this PIITYPE. The names listed in this element are matched as substrings against the column names in the database. This means that an ALIAS 'TITLE' also matches 'SUBTITLE', 'TITLESTRING' columns.

The other elements part of this PIITYPE definition are explained in more detail in the next chapter and won't be detailed here.

```xml
<PIITYPES>
  <PIITYPE name="Title of address or honorific" type="list" class="PII" threshold="20" risklevel="1" mask="FPEtitle" enabled="1" id="1">
    <TRANSLATIONS>
      <TRANSLATION name="Title of address or honorific"  language="en" />
      <TRANSLATION name="Titre"                          language="fr" />
      <TRANSLATION name="Titolo Onorifico"               language="it" />
    </TRANSLATIONS>
    <ALIASES>
      <ALIAS name="TITLE" />
    </ALIASES>
    <DATAFILES>
      <DATAFILE name="Brazil Titles"     file="title.br.txt" locale="br" enabled="0" img="flags/br.png" />
      <DATAFILE name="Belgium Titles"    file="title.be.txt" locale="be" enabled="0" img="flags/be.png" />
      <DATAFILE name="German Titles"     file="title.de.txt" locale="de" enabled="0" img="flags/de.png" />
      <DATAFILE name="French Titles"     file="title.fr.txt" locale="fr" enabled="0" img="flags/fr.png" />
      <DATAFILE name="Italian Titles"    file="title.it.txt" locale="it" enabled="0" img="flags/it.png" />
      <DATAFILE name="Portuguese Titles" file="title.pt.txt" locale="pt" enabled="0" img="flags/pt.png" />
      <DATAFILE name="Turkish Titles"    file="title.tr.txt" locale="tr" enabled="0" img="flags/tr.png" />
      <DATAFILE name="Spanish Titles"    file="title.es.txt" locale="es" enabled="0" img="flags/es.png" />
      <DATAFILE name="English Titles"    file="title.uk.txt" locale="gb" enabled="0" img="flags/gb.png" />
      <DATAFILE name="US English Titles" file="title.us.txt" locale="us" enabled="1" img="flags/us.png" />
    </DATAFILES>
  </PIITYPE>
</PIITYPE>
```

Different techniques to define a grammar

There are different ways to define a grammar and its rules. This chapter will walk through the different techniques and provide an example for each of them. The PII-def.xml provides the structure of how the rules should look like and what information needs to be added to add additional types. Examples from the PII-def.xml file will be used to explain the different techniques.

Regex

A regular expression can be used to match a pattern against the values in a column in the source database.

A sample of a definition of a grammar type with regular expressions can be found in the image below. The following points are important when defining a grammar of a regex type.

- In the PIITYPE, the type should be defined as: regex

- The tag <PATTERNS> marks the start of all the patterns that are defined within this grammar

- Each <PATTERN> contains a single pattern definition, it contains a:

    ○ Name – name of the rule in the UI

    ○ Regex – the regular expression that is used to match values

    ○ Locale – Optional - to indicate which country the grammar is valid for

    ○ Enabled – 0 or 1, 0 means disabled, 1 means that the rule is enabled

    ○ Img – Image that is shown in the UI with this rule (a little flag)

    ○ groovyFile – optional – to indicate which script file should be executed when matching this rule

    ○ methodName – optional – indicates which method / function to execute when matching this

rule

A sample of a regular expression that was created for IT-CE is:

```
<PATTERN name="French SSN With Checksum"

regex="\b[1-478][ ]?\d{2}[ ]?\d{2}[ ]?[0-9][0-9AB][ ]?\d{3}[ ]?\d{3}[ ]?\d{2}\b"

locale="fr" enabled="1" img="flags/fr.png"

groovyfile="validate" methodname="isFrenchSSN" />
```

To test and validate regular expressions an online website can be used: https://regex101.com/

Dictionary

A dictionary can be used to match a list of values against the values in a column in the source database.

A sample of a definition of a grammar type with dictionaries can be found in the image below. The following points are important when defining a grammar of a dictionary type.

- In the PIITYPE, the type should be defined as: list

- The tag <DATAFILES> marks the start of all the dictionaries that are defined within this grammar

- Each <DATAFILE> contains a single pattern definition, it contains a:

  ○ Name – name of the rule in the UI

  ○ File – the file that contains the list of values that need to be matched

  ○ Locale – Optional - to indicate which country the grammar is valid for

  ○ Enabled – 0 or 1, 0 means disabled, 1 means that the rule is enabled

  ○ Img – Image that is shown in the UI with this rule (a little flag)

- The dictionary file MUST be saved in UTF8

```
<PIITYPE name="Postal Code List" type="list" class="CONTACT" threshold="90" risklevel="2" mask="FPEzipcode2" enab
  <TRANSLATIONS>
    <TRANSLATION name="Postal Code"   language="en" />
    <TRANSLATION name="Code Postal"   language="fr" />
  </TRANSLATIONS>
  <ALIASES>
    <ALIAS name="POSTCODE" />
    <ALIAS name="POST_CODE" />
    <ALIAS name="PCODE" />
    <ALIAS name="POSTAL" />
    <ALIAS name="POSTALCODE" />
    <ALIAS name="DISTRICT" />
    <ALIAS name="ZIPCODE" />
    <ALIAS name="ZIP" />
  </ALIASES>
  <DATAFILES>
    <DATAFILE name="French Postal Code"      file="postalcodes.fr.txt"        locale="fr" enabled="1" img="flags/f
  </DATAFILES>
</PIITYPE>
```

A sample of a dictionary that was created for IT-CE is postalcodes.fr.txt:

```
24000
24001
24002
24003
24004
24005
24007
24009
24010
24011
24012
24013
24014
24015
```

Date Range

A date range can be used to validate a range in years against a column (type date) in the source database.

A sample of a definition of a grammar type with a date range can be found in the image below. The following points are important when defining a grammar of a range type.

- In the PIITYPE, the type should be defined as: range-date

- The tag <RANGE> defines the range that is used and contains the following:

  ○ Min – minimum amount of years difference

  ○ Max – maximum amount of years difference

  ○ Enabled – 0 or 1, 0 means disabled, 1 means that the rule is enabled

○ Exclude – database columns that contain these substrings will be excluded from matching

```
<PIITYPE name="Date of Birth" type="range-date" class="PII" threshold="80" risklevel="2" mask="FPEDOB"
  <TRANSLATIONS>
    <TRANSLATION name="Date of Birth"   language="en" />
    <TRANSLATION name="Date de Naissance"   language="fr" />
  </TRANSLATIONS>
  <ALIASES>
    <ALIAS name="BIRTH" />
    <ALIAS name="BIRTHDAY" />
    <ALIAS name="BIRTHDATE" />
    <ALIAS name="DOB" />
    <ALIAS name="DATEOFBIRTH" />
    <ALIAS name="DATE_OF_BIRTH" />
  </ALIASES>
  <RANGE min="25" max="80" enabled="1" exclude="upd,cre,maj" />
</PIITYPE>
```

Number Range

A number range can be used to validate a range of numbers against a column (type numeric) in the source database.

A sample of a definition of a grammar type with a number range can be found in the image below. The following points are important when defining a grammar of a number range type.

- In the PIITYPE, the type should be defined as: range-number

- The tag <RANGE> defines the range that is used and contains the following:

    ○ Min – minimum number

    ○ Max – maximum number

    ○ Enabled – 0 or 1, 0 means disabled, 1 means that the rule is enabled

```
<PIITYPE name="Age" type="range-number" class="PII" threshold="80" risklevel="1" mask="FPEage" enabled="0" id="6">
  <TRANSLATIONS>
    <TRANSLATION name="Age"   language="en" />
    <TRANSLATION name="Age"   language="fr" />
  </TRANSLATIONS>
  <ALIASES>
    <ALIAS name="AGE" />
  </ALIASES>
  <RANGE min="25" max="80" enabled="1" />
</PIITYPE>
```

String Range

A string range can be used to validate the length of the text in a column (type text/varchar) in the source database. It can be useful to identify columns that contain a lot of text

A sample of a definition of a grammar type with a string type can be found in the image below. The following points are important when defining a grammar of a string type.

- In the PIITYPE, the type should be defined as: range-string

- The tag <RANGE> defines the range that is used and contains the following:

    ○ Min – minimum length of the text

    ○ Max – maximum length of the text

○ Enabled – 0 or 1, 0 means disabled, 1 means that the rule is enabled

```
<PIITYPE name="Free Text Note Field" type="range-string" class="TEXT" threshold="20" risklevel="1" mask="FPEfreetext">
  <TRANSLATIONS>
    <TRANSLATION name="Free Text Note Field"   language="en" />
    <TRANSLATION name="Texte Libre"   language="fr" />
  </TRANSLATIONS>
  <ALIASES>
    <ALIAS name="NOTE" />
    <ALIAS name="NOTES" />
    <ALIAS name="REMARK" />
    <ALIAS name="REMARKS" />
    <ALIAS name="COMMENT" />
    <ALIAS name="COMMENTS" />
    <ALIAS name="DETAIL" />
  </ALIASES>
  <RANGE min="20" max="10000" enabled="1" />
</PIITYPE>
```

Validation Functions

Validation functions can be used in case the values in the database that are being matched have additional information that can be used to validate whether the value is valid. An example is a credit card number that contains control digits at the end of the number. A validation function can be used to calculate and validate that the control digits are valid for the credit card number that was identified using the grammar.

Validation functions are stored in the `validate.groovy` file and as the filename indicates are written in *groovy* programming language.

The original `validate.groovy.example` file can be renamed to `validate.groovy` to make all the functions active. To perform a function for a grammar this needs to be configured with the attributes:

- groovyFile – to indicate which script file should be executed when matching this rule

- methodName – indicates which method / function to execute when matching this rule

An example of a validation function is the function used to validate a French SSN 'Social Security Number) as displayed in the image below:

```
// French SSN
// This validation logic is based on information available at:
//     https://fr.wikipedia.org/wiki/Numéro_de_sécurité_sociale_en_France
boolean isFrenchSSN(String input) {
    String id = input.replaceAll(/([~!@#$%^&*()_+=`{}\[\]\-|\\:;'<>,.\/? ])+/, '')
    id = id.replaceAll(/(ABab)+/, '0')
    if (id.size() != 15) {
        return false
    }
    long idWithoutCheckDigits = Long.parseLong(id.substring(0, 13))
    long checkDigits = Long.parseLong(id.substring(13, 15))
    long rem = 97 - (idWithoutCheckDigits % 97)
    if (rem == 0) {
        rem = 97
    }
    return checkDigits == rem
}
```

Only values that match the regular expression AND for which the function returns 'true' are counted as positive hits.

To install to test GROOVY functions, download a windows installer version of Groovy (e.g., Groovy 2.4.17) from: http://groovy-lang.org/download.html

After installing GROOVY (requires JRE install as well) you can test and validate the validation functions that you write for the SDM solution.

Best practices

The following best practices apply when defining grammars

1. **Only 1 type of rule per PIITYPE**

   You can only define a single 'type' of grammar for a PIITYPE, this means that your PIITYPE can not contain regular expressions and dictionary types at the same time. If you want both types of definitions two PIITYPE's need to be configured.

2. **Aliases**
   An unlimited number of ALIASES can be defined for a PIITYPE, however keep in mind that the ALIASES are matched against the column names using 'substring'. This means that if you define an 'ALIAS' named 'CREDITCARD' it also matches columns named 'CCREDITCARD', 'CREDITCARD_2' etc. This also means that ALIASES should be chosen wisely, they shouldn't be TOO SHORT (e.g., 'CC') because it would match every column that has a name containing two C's.

3. **Accent sensitivity**
   Dictionaries are not accent sensitive; this means that values in the dictionary that are listed with an accent also match values in the database without an accent.

4. **Configuring the weight of a grammar**
   The weight of a grammar can be configured by configuring a set of values in the PII-def.xml file for a PIITYPE. The higher the weight, the higher the risk displayed on the dashboard when a column matches the PITTYPE. The parameters below can be found in the PII-def.xml file and have already been mentioned in the earlier chapters, therefore only a reference to them will be listed here. The following configuration parameters determine the risk / weight for a column that matches:

   - PIICLASS weight: The higher the weight of the CLASS a grammar belongs to, the higher the calculated risk

   - PIITYPE risklevel: The higher the risklevel of a PIITYPE, the higher the calculated risk

   The higher the percentage of matches in the source database column, the higher the calculated risk will be using the two configuration values above.

   It is important to define the grammars that are most important to the customer with the highest weights, this way the most important matches in the source databases get the highest risk associated and can be reviewed first.

5. **Reducing false positives**
   When matching against grammars it's important to reduce the number of false positives. This can be achieved by making sure your regular expressions are as specific as possible, but also

by using the threshold configuration option to lower the likeliness that a regex that matches a lot of false positives is marked as a risk.

The following configuration parameters can be used to reduce the number of false positives:

- PIITYPE threshold: The threshold indicates the percentage of values in the source database column that at least should match before the PIITYPE is marked as relevant. If the database contains 100 rows and the threshold is configured as 30, at least 30 rows must match the PIITYPE for the PIITYPE to be indicated as a risk. It's important to keep the threshold low for grammars that are not very specific and keep the threshold high for grammars that are more specific.

- Validation functions: Validation functions can be used on top of a regular expression to validate that a number found is valid for this PIITYPE (e.g., is it a valid credit card number or not). This should be used to help reduce the number of false positives where possible.

Grammar classes

Classes are classification of grammar types. For example, personal, health, financial, and contact information.

Grammar classes are categories of Grammar types. They can group all financial types, or all contact types for example. Grammar classes have a weight. This weight will be used to calculate the risk score.

Grammar types and rules are collected in Sets. Discovery module is installed with the Global Set that includes all types and rules. This set is called PII-def and will be the parent for any set you would like to create

- It is recommended not to have more than 7 classes.

- The following classes are defined up front: Financial, Personal, Contact, Health, Other

- Weight will be used in the formula that calculates the risk score.

Create new class

Button/Field	Description
New Class	1. Navigate to the **Grammar** tab in the Discovery module. 2. Click **New Class**. The **Grammar Class (New)** dialog is displayed. 3. Enter the **Code**, **Name**, **Description** and **Weight** between 0 to 10. 4. Click **Save** to add the new class.

Edit a class

Button/Field	Description
Edit	1. Click ✏️ icon for the class you want to edit. The grammar classes (update) dialog is displayed. The only fields you can update are **Name**, **Description**, and **Weight**. 2. Updated the information and click **Save**.

Grammar set

Sets collection of types. You can assign a set to a project based on the country or locale. In the project, you can link a particular set with the project. Once you define sets, you can tune the rules or types as per your requirement.

Button/Field	Description
New Set	1. Click . The Grammar Set (New) dialog box is displayed. 2. Enter the **Code**, **Name** and **Description**. By default, PII-def is selected as parent which cannot be changed. 3. Select the check box against the country flags. All the rules associated with the selected country(s) will be added in the new set. 4. Click **Save**. The newly created set will be listed in Grammar sets page. Click the code in the sets to view the Grammar set rules.
Edit set	1. Click icon for the set you want to edit. The grammar sets (update) dialog is displayed. 2. Select the check box against the country flag and update the set as per your requirement. 3. Click **Save**.
Publish a set	For the newly created grammar set to be operational, you need to publish the set. When you publish a set, **<set name>.xml** is generated in `<install_dir>\config\obtdiscovery` To publish a set, navigate to grammar set page and click icon.

| View as XML | You can view the .xml of a published set.
| | 1. Click icon to view the set as XML.
| | The XML for the set is displayed in a dialog.
| | 2. Click **Cancel** to exit out of the XML. |

Grammar set rules

This page consists of information about the set and its rules.

Grammar Set Rules

Dashboard   Sources   Projects   **Grammars**   Monitor   Schedule   Configuration   Search   Logs

DASHBOARD   CLASSES   SETS   **SET RULES**   TYPES   MASKING   GROOVY

Grammar Set	**Global Grammar**		ID	PII-def
Description	Global Grammar		Parent	
Filter	WW ALL			

● ON (35)   ● OFF (270)   ● WIP

Show 10 ▾ rows

Search...

NAME	TYPE	LOCALE	RULE		
Nationality	list	gb	datafile	nationality.en.txt (UTF-8)	●
Country	list	us	datafile	countries.ww.txt (UTF-8)	●
Family Name	list	us	datafile	lastnames.us.txt (UTF-8)	●
Firstname	list	us	datafile	firstnames.us.txt (UTF-8)	●
Gender	regex	us	pattern	^(male|man|M|female|woman|F)$	●
Identifier: Driving License	regex	us	pattern	\b[A-Za-z][2] ⊕ **validate.isUSdrivingLicense()**	●
Identifier: Health	regex	us	pattern	\b((\d){3})(-)?(\d)[2](-)?(\d)[4](A|B[1-7]?|M|T|C[1-4]|D)\b	●

The page displays the status of number of rules that are in **ON**, **OFF** and **WIP** state. The details below the status include name, type, locale, rule, rule description and status. Click the circle in the status

column against a rule to enable or turn it off or move it to WIP. Click **Save** to reflect the changes in status of the rules.

Click the country flags or icons to display all the rules assigned to that. This feature helps you to act only on those rules associated with that country.

Grammar types

Types helps you define sensitive information, like, social security number, credit card number, family name, passport name and so on.

Name?	Description?	Type?	Class?	Threshold?	Risk Level?	ENABLED	IDENT	Masking?	Aliases?	
Bank Account	Bank Account	regex	PCI	20	5	☑	☑	FPEbankaccount	BANK,BANKACCOUNT,BANK_ACCOUNT,ACCOUNTNUMBER,ACCOUNT_NUMBER	✏
Birth Year	Birth Year	range-number	PII	40	1	☑	☑	FPEyear	BIRTH,YEAR	✏
City	City	list	CONTACT	20	2	☑	☑	FPEcity	CITY,TOWN	✏
Country	Country	list	CONTACT	20	2	☑	☑	FPEcountry	REGION,COUNTRY	✏
Credit Card	Credit Card	regex	PCI	20	5	☑	☑	FPEcc	CREDIT,CCNUMBER,CARD,CREDITCARD	✏
Date of Birth	Date of Birth	range-date	PII	40	2	☑	☑	FPEDOB	BIRTH,BIRTHDAY,BIRTHDATE,DOB,DATEOFBIRTH,DATE_OF_BIRTH	✏
Email Address	Email Address	regex	CONTACT	20	3	☑	☑	FPEemail	EMAIL,EMAILADDRESS,EMAIL_ADDRESS	✏ 🗑
Family Name	Family Name	list	PII	20	3	☑	☑	FPEfamilyname	SURNAME,LASTNAME,LNAME,FAMILYNAME	✏
File Path	File Path	regex	OTHER	20	1	☑	☑	FPEfilepath	PATH,FILE	✏
Firstname	Firstname	list	PII	20	3	☑	☑	FPEfirstname	FIRSTNAME,FIRST_NAME,GIVENNAME,FNAME	✏

Showing 1 to 10 of 27 rows

DataTypes	Note	Origin	SharedWith	MinSize	MaxSize	Zap	MapTo
string				1	-1		

You can define a type with one of the following data types:

- Regular expression

- New dictionary (list)

- Method

- Date range

- Number range

- Word count range

To view the grammar rules for a particular type, click on the name of the type. See Grammar Rules.

Add new type

Button/Field	Description
New Grammar Type	1. Depending on the data type, click the icon to add new type. The Grammar Types dialog is displayed. 2. Enter **Name** and **Description**. The name is the unique identifier when defining a type. 3. Select the **Class** from the drop down. 4. Move the slider to set the **Threshold**(0 - 100) and **Risk Level** (0 - 10). 5. Check the check box to enabled or disabled. 6. If you have a masking function to be executed for this type, select **Masking** from the drop down. 7. Enter **Note**, **Origin**, **Shared** and **Aliases**. You can specify comments in note. In Origin, you can mention from where the information is originating or coming from, is it data entry or some external feed. In Shared, you can mention whether you are sharing this type of information with other organization or external party for business purposes. The Origin and Shared information help to access the risk involved with the type. If you have multiple aliases, separate them with a comma (,). 8. Click **Save** to add a new type.

Edit a type

Button/Field	Description
Edit Type	1. Click type you want to edit. The grammar types (update) dialog is displayed. 2. Modify or update the details and click **Save**.

Grammar Rules

Since a rule is associated with a type you can access rules only from a particular type. Click on the name of the type to display rules linked to that type.

Depending on the data type of a type, you can define rules with a regular expression, groovy script pointing to a method, parameter, list, or a range.
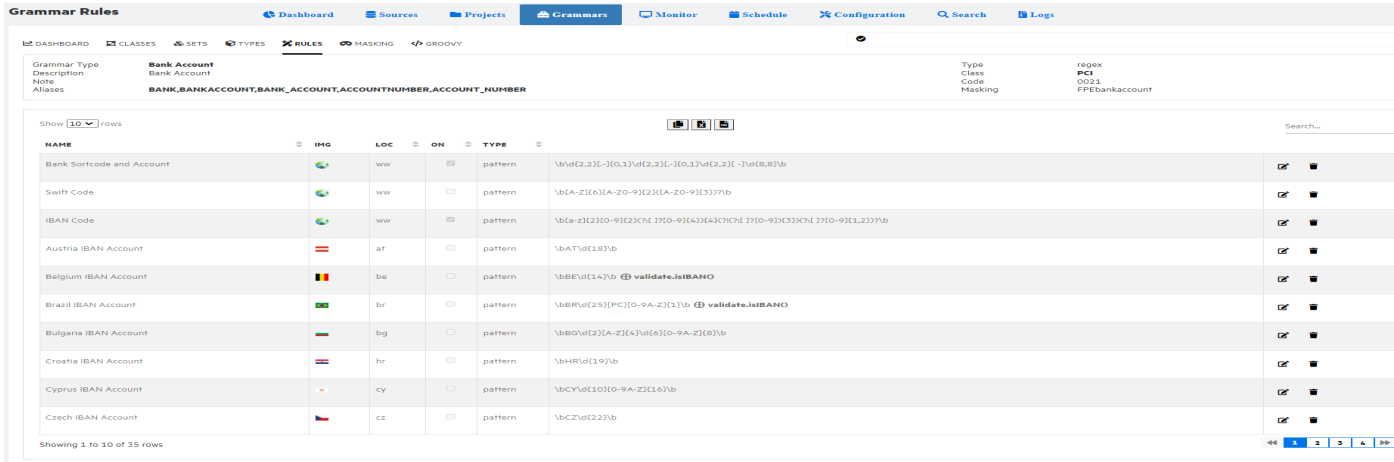
The rules list the name, locale, check box if the rule is enabled or not, type, and description of the type. You can have one or more rules linked to one type. You can enable rules as per requirement. For example, a credit card can have rules to validate the numbering pattern and methods to validate if the number entered is credit card or fake and so on.

Add new rule

To add new rule, perform the following steps:

Button/Field	Description
New Rule	1. Click **Add** icon (File, table, method, or regular expression). The Grammar rules dialog is displayed. **Name** is auto generated unique identifier. 2. Enter **Description**. 3. You can use image file to represent the rule. 4. Select **Locale** from the drop down. 5. Check the check box to enable rule. 6. Enter regular expression in **Regex** or link a groovy script in **Groovy** with a **Method** or specify a parameter in **Param** or provide a text file listing the value or description in **File**. 7. Enter any comments in Notes. 8. Click **Save**.

Grammar masking strategies

Masking strategies are how you would like to act on the data for each Grammar type. A masking strategy will be assigned to a Grammar type. This page lists the masking functions defined using groovy.

It includes ID, code, use, prod, test, Dev, QA, and education. You can define how a type can be tuned or told to function based on the phase on which this system is used whether it is production, test, dev, QA, or education.

and displayed on the top right refers to different businesses an organization has. For example, an organization can have banking as well as insurance business. In this case, can represent banking and can represent insurance and you can define masking functionality based on this.

With the masking strategies, you can further process or act on the data using Remediation option.

Grammar groovy

This page lists the Validate.groovy and List.groovy scripts. Click on the validate or list groovy script to view the content. If you want to modify the script, edit as needed and click **Save validate** or **Save List**.

Groovy

Dashboard | Sources | Projects | **Grammars** | Monitor | Schedule | Configuration | Search | Logs

DASHBOARD | CLASSES | SETS | TYPES | MASKING | GROOVY

☑ SAVE VALIDATE | SAVE LIST

☑ validate.groovy

list.groovy

Monitor

This page displays details of ongoing scans and the last scans completed on a table cross project level. If an ongoing scan is taking long time or not progressing as expected, then you can stop the scan.

Button/Field	Description
All	List all the project tables which are scanned, aborted, and erred
Done	List all the project tables which are scanned successfully
Abort	List all the project tables which aborted during the scan
Error	List all the project tables which erred during scan
Scanning	List the project tables for which scanning is in progress

Schedule

This page displays details of the scheduled tasks.

> **NOTE:** Make sure the Discovery service is installed before you start it. To install discovery service, see Discovery Service installation

The following tabs in schedule are explained below:

- Scheduler
- Service commands
- Command log
- Service log

Scheduler

Lists all the jobs waiting to be processed along with other details, such as, project name, name, username, next run, cron command, scan level, enabled, cws, and regnode

- Select a schedule,
- Click **Actions** drop-down and select **Change status** to enable or disable the schedule.
- Similarly, click the **Exec on** from the **Action** drop-down to start or stop the discovery service.

The CWS column tells you who is responsible for processing the job. The CWS represents three executors, namely, Command line, Web console, and Service. They have the following unique identification numbers:

1 - Command line

2 - Web Console

3 - Service

- **WebConsole Registered Jobs** : Display the jobs which are scheduled to run.

- **Current Discovery Service** : Indicates the service id of the current running discovery service.

  If you have more than one service running, you can change the current service by selecting the service id from the service column.

- **Discovery Service Jobs**: Indicates the jobs which are scheduled to run on the current running discovery service

WebConsole Registered Jobs
Discovery Service
Discovery Service Jobs

| 16.103.3.195 - Ash-Win - Windows Server 2019 |
| analyzetables | refreshmv |

Service	Node	Jobs
3	16.103.3.195 - Ash-Win - Windows Server 2019	[[analyzetables, DiscoveryGroup2, Wed Jan 19 22:00:00 WGT 2022], [refreshmv, DiscoveryGroup2, Wed Jan 19 23:00:00 WGT 2022]]
6	16.103.3.195 - Ash-Win - Windows Server 2019	[[scan_project_2_table_LIST_00002, DiscoveryGroup1, Thu Jan 27 04:15:00 WGT 2022]]
9	16.103.3.195 - Ash-Win - Windows Server 2019	[[scan_project_2_table_LIST_00003, DiscoveryGroup1, Mon Jan 31 04:17:00 WGT 2022]]

Service commands

Allows you to communicate directly with the service. The service can execute commands on demand or execute scheduled jobs. You can enter the commands and directly requests the service to process the job. There are three different default templates with commands that you can make use of to communicate with the service.

Schedule   Dashboard   Sources   Projects   Grammars   Monitor   **Schedule**   Configuration   Search   Logs

SCHEDULER   SERVICE COMMANDS   COMMAND LOG   SERVICE LOG

SEND TO SERVICE (3)

show projects;
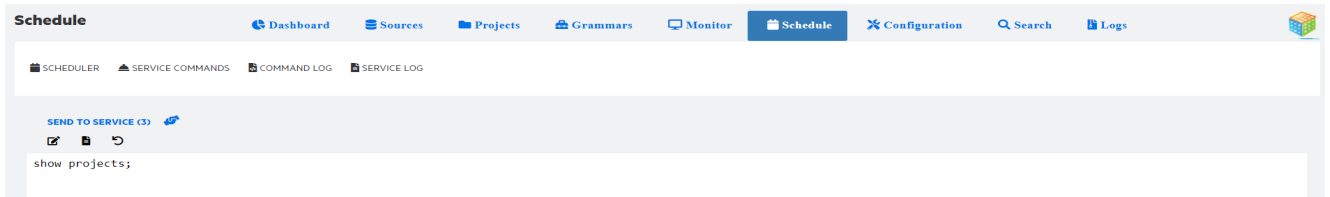
Button / Field	Description
Template 1 and **Template 2**	Currently, there are two templates available for ease of use.
Language Help	Click icon to display the SDM discovery language help page.
SEND TO SERVICE	Click to send the commands to the currently running service

Command log

Displays the results of the commands executed.

Service log

Once the discovery service is started, this tab displays the service logs. For example, service started, registered jobs, execute commands and so on. At regular intervals, the service checks if it has any jobs to process. By default, the interval is set to 2 minutes. All these events are logged here.

Configuration

This page allows you to configure scanner, UI, view data, scan blackouts, JDBC, sampling, registrations, and others.

Button /Field	Description
Scanner	Allows you to enable location support, logging, minimum sampling, minimum percentage of sampling, grammar threshold, and field not null count. • **Enable Location Support**: Enable user defined locations to be source database for scanning. If disabled only the OLTP_DB location can be scanned. • **Logging**: Add an entry in the log file after every n sampled rows • **Minimum Sample**: Minimum number of rows to sample. The scanner will scan the greater of samplesize and samplepercent up to samplemax rows. • **Maximum Sample**: Maximum number of rows to sample. The scanner will scan the greater of samplesize and samplepercent up to samplemax rows.

	• **Grammar Threshold**: Hit ratio threshold for grammar to be classified as a candidate, over-ridden by project grammar thresholds. • **Field NOT NULL Count**: Perform a full table count of not null values for each column scanned.
UI	Allows you to choose language, project views, item limit, data tables rows and project page. • **Language**: Language of the user interface. • **Project View**: Select default view of project list. • **Item Limit**: Maximum number of items to show in tables, larger number can affect UI performance. • **DataTables Rows**: Data tables rows to show • **Project page**: Preferred project page • **Register Tables Limit**: Allows you to set maximum number of tables that can be registered at a time.
View Data	• **Allow Data View**: Allows you to enable users to view column data • **Sample Size**: Sets maximum number of rows visible on a page. **NOTE:** Make sure that the user belongs to a group with **View data in discovery** role. For more information see, Roles and Privileges
Scan blackouts	Specify day and time where no scanning should take place. You can have scan blackout globally or based on the database. You can click **Set no constraints** icons and choose blackout time from 6-18, 7-17, 8-18, 18-23, 19-23and 20-23.
JDBC	Lists all the JDBC drivers for different databases. For example, Hive, Cassandra, ODBC, JTDS etc.
Sampling	Displays the SELECT queries for different databases used for the sample during scanning.
Registration	Lists the registered scan types like tables or views for different databases.
Indexing	Displays the install path of Solr, download directories, Solr url, and Web Console url parameter values used for indexing.
Metadata	Displays sample SELECT queries of metadata.
Counting	Displays the sample SELECT queries for counting on different databases.

DT mapping	XML file for data type mapping.

Icons

Icon	Description
◔	Analyze discovery tables
⟳	Refresh discovery materialized views
ᴉᴉᴉ	Refresh Discovery statistics

Search

This page allows you to search a particular PII in a project, table, or column. You can search using **AND** conditions.

> **NOTE:** Make sure that Solr is installed and running for the search to operate.

- [Solr installation](#)
- [Solr configuration](#)
- [Search a PII](#)

Solr installation

To install Solr from discovery, perform the below steps:

1. Launch command prompt under the

   ```
   >cd <Install_dir>\obt\bin
   ```

```
Discovery.bat -a
```

2. Run the following commands

   a. `install solr` : wait for the solr installation to finish, and after installation the Solr automatically starts on port 8983.

   b. `create solr` : creates the solr core **sdmdiscovery**

   c. `check solr`: checks the status of Solr

   d. `refreshmv`: refreshes the materialized view

   e. `index solr`:creates the indexes on Solr based on the project, tables, and columns in discovery

Solr configuration

If you already have a Solr indexing server running, you need to configure Discovery to point it to that Solr. Follow below steps to configure discovery:

1. On Windows

   ```
   > cd <Install_dir>\obt\bin
   ```

   ```
   Discovery.bat -a
   ```

2. `set param solrurl=Enter the solr URL`

   For example, `set param solrurl=http://<hostname>:<solrport>/solr`

3. In the command prompt, type `q` and press **Enter**.

   Parameters are cached. So, if you change the parameters you need to perform this step to update the values

4. Run `Discovery.bat -a`

5. Run `show param solrurl` to verify the solrurl is updated with the value you specified.

6. Make sure you create a core sdmdiscovery on Solr or set the Solr core parameter value using below step:

   ```
   set param solrcore=<corename>
   ```

   For example, `set param solrcore=corename`

   Quit and relaunch discovery command prompt using step 3 and 4.

7. `index solr`

   This command creates the indexes on Solr based on the project, tables, and columns in discovery

> **NOTE:**
> - For Solr install or configuration on Linux run the same commands by launching the discovery command line using `sh Discovery.sh` or `./Discovery.sh`
>
> - For more Solr commands refer solr Commands.

Search a PII

To search a PII from UI, perform the following search:

1. Enter or select the PII from the drop down. You can also use the And condition and type the PII combination.

2. Click **Projects**, **Tables** or **Columns**. The result set in displayed on the page with link to the location, type and solr score.

3. You can also click **All** to search in **Projects**, **Tables**, and **Columns** all at once.

> **NOTE:** If you want to search a PII from command line then refer to the commands specified under Solr commands in solr Commands, on page 86.

Logs

This page displays logs generated for scans, when a reset was done, command line logs, SDM and Tomcat logs. The logs contain information like, ID, directory, name of the log file, size, and date.

- **SCAN** : Lists all scan logs across all projects

- **RESET**: Lists all reset logs across all projects

- **CMDLINE**: Lists all the command line output logs

- **SDM**: Lists all the sdm logs such as obt.log, environment logs, job run logs from OBTHOME

- **TOMCAT**: Lists all the tomcat and webConsole service logs

Button /Field	Description
Process	To view the logs, perform the following steps: 1. Click the check box for the logs you want to view. 2. Click **Process** on the top right corner of the page. The Process log files dialog is displayed. 3. Click **View**. The log file is displayed as pop up. Click **Delete** will delete the log.

Command Line Interface

This section describes, how to install and use Discovery using command line arguments.

- [Introduction](#)
- [CLI commands](#)

Introduction

SDM Discovery module offers a simple interface to enable running Discovery commands in a batch mode.

On Windows

Navigate to `<Install_dir>\obt\bin` and type `Discovery.bat`

> **NOTE:**
> - On Windows ANSI console is not installed by default
> - You always need to run with `Discovery.bat -a`

- Command line arguments

  usage: `Discovery.bat -h [-a] [-f filename] [-d directory] [-k key]`

On Linux

Navigate to `<Install_dir>/obt/bin` and type `./Discovery.sh`

> **NOTE:**
> - On Linux ANSI console is installed by default
> - You need not run with the `-a` option.

- Command line arguments

  usage: `Discovery.sh -h [-a] [-f filename] [-d directory] [-k key]`

-a,--ansi	Initialize ansi console.
-d,--directory directory	Directory name.
-f,--filename filename	File name.
-h,--help	Usage information.
-k,--key key	Encryption key.

batch mode - start one file

- `Discovery.bat  f commandfile -d commanddir -k key`

Interactive mode

- `Discovery.bat  d commanddir -k key`

- `Discovery.bat  k key`

- `Discovery.bat`

> **TIP:** `key` is SDM encryption key that allows access to SDM repository.

if `-k` is not used, you will be prompted for the key.

Type `h` for help.

Type `q` to quit.

CLI commands

> **IMPORTANT:**
>
> - Set project commands must be issued before create project or update project.
>
> - Set grammar commands must be issued before add grammar.
>
> For example, see the below figure showing the flow.
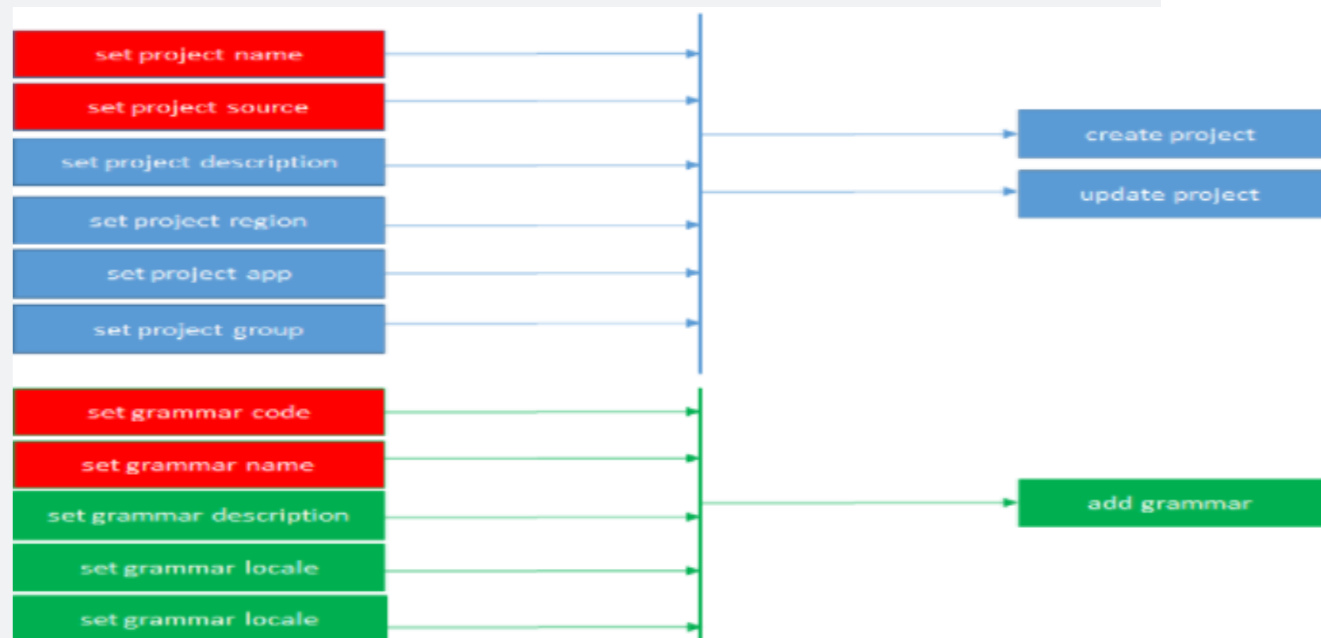
The following CLI commands are used in Discovery. For detailed examples on the usage of the CLI commands see, Language Help

- set Commands

- params Commands

- show Commands

- grammar Commands

- set project Commands

- project Commands

- solr Commands

- scan Commands

- other Commands

- masking Commands

- service Commands

- datasource Commands

set Commands

set commands enable to set prerequisite parameters for subsequent commands.

set command dir	set command directory	set debug
set fuzzylist	set log dir	set log directory
set log file	set output dir	set output directory
set output file	set pgdebug	set threads
set user		

Example:

```
set project name
```

Sets the name before a create project or an update project

params Commands

This section describes **params** commands.

delete param	delete parameter	forcedelete param
forcedelete parameter	set param	set parameter
show param	show parameter	show parameters
show params		

Examples:

```
show params %db2%;

set param logRows=1000;

delete param xxx;
```

show Commands

show commands enable to list Discovery objects.

show cd	show command dir	show command directory
show command file	show context	show cron
show debug	show env	show errors
show log dir	show log directory	show log file
show messages	show normalized name	show output dir
show output directory	show output file	show pgdebug
show schedule	show schedules	show threadmethod
show threads	show user	show valid name

Example:

```
show current project
```

Shows the content of current project

grammar Commands

This section describes **grammar** commands.

add grammar	add grammar redo	load grammar
publish grammar	publish htmlgrammar	publish minigrammar
publish minihtmlgrammar	set grammar code	set grammar desc
set grammar locale	set grammar name	set grammar parent
show gclasses	show grammar	show grammar code
show grammar desc	show grammar locale	show grammar name
show grammar parent	show grammar stats	show grammars
show gtypes		

Examples:

```
check dic;

publish grammar;

publish htmlgrammar;
```

set project Commands

This section describes **set project** commands like:

set current project	set project	set project app
set project cmclass	set project cmloc	set project cmrtype
set project countnotnull	set project countsql	set project countusql
set project dashboard	set project desc	set project description
set project domain	set project email	set project grammar
set project group	set project lock	set project logo
set project name	set project noscantimewindow	set project region
set project remotefile	set project reservedby	set project samplemax
set project samplepercent	set project samplesize	set project samplesql
set project scandbtypes	set project service	set project source
set project sourcetype	set project tags	set project threads
set project type	set project undashboard	set project unlock
set table	set tables note	set tables tags

Examples:

```
set project desc;

set project group;

set project app;
```

project Commands

This section describes **project** commands.

count schema	count schemas	create project
delete project	delete project tables	delete tables

export project	import project	load project
register deltatables	register project deltatables	register project schemas
register project table	register project tables	register schemas
register table	register tables	review project
review table	split project	update project
validate name	validate project name	

> **NOTE:**
> - By default, the maximum number of tables that can be registered at a time is 5000.
> - Use `set param maxRegisterTables=<value>;` to change the default value.

Examples:

```
set param maxRegisterTables=10000;

register schemas xxxx;

register tables all;

delete project;
```

show project Commands

This section describes show project commands.

show audit	show columns	show columns noalias
show columns stats	show columns withalias	show deltaschemas
show deltaschemas	show info	show noscantimewindow
show project audit	show project deltaschemas	show project deltatables
show project grammar	show project info	show project noscantimewindow
show project schedule	show project schedules	show project schemas
show project stats	show project tables	show projects
show schemas	show stats	show tables

Examples:

```
show info;

show tables;
```

```
show deltatables;
```

solr Commands

solr commands as a set of command to manage indexing of SDM Discovery repository content.

> **NOTE:** Parameters are cached. So, if you change the following parameters.
>
> - `solrurl`
> - `wcurl`
>
> required
>
> - `exit discovery.bat/.sh`
> - `deleteall solr`
> - `refresh mv`
> - `index solr`
>
> This refreshes the index with the new values of `wcurl`

check solr	create solr	deleteall solr
index solr	install solr	restart solr
search project	search solr	set search all
set search column	set search level	set search project
set search query	set search table	start solr
stop solr		

scan Commands

This section describes **scan** commands.

deepreset project scan	deepreset scan	deepreset table
deepreset table scan	deepreset tables	deepreset tables scan
reset project scan	reset scan	reset table
reset table scan	reset table scan	reset tables scan
scan project	scan project table	scan project tables
scan table	scan tables	show project scan
show scan	show scanall	

Examples:

```
scan project;

reset project scan;

scan tables all;
```

other Commands

This section describes **other** commands.

analyze tables	autocreate projects	check dic
check dictionaries	check install	check messages
convert dics	convert messages	correct override
correct project override	delete project schema	delete schema
generate envcsvdml	generate envdml	generate images
import column desc	import column descriptions	import table desc
import table descriptions	load snapshot	pause refresh stats
refresh fieldmap	refresh mv	refresh stat
refresh stats	refreshmv	register schedule
report dic	report dictionaries	resume refresh stats
schedule script	set max tables	set single schema
show max tables	show single schema	show split
show version	snapshot	start
top columns	top projects	top tables
unschedule job	validate install	

Examples:

```
import table desc;

refresh stats;

start xxx;
```

masking Commands

This section describes **masking** commands.

generate compositions	generate compositions force	generate designer
generate masking	generate shuffle sql	generate shuffles
generate shuffles force		

Examples:

```
generate masking;

generate shuffles sql;

generate shuffles;
```

service Commands

This section describes **service** commands.

service delete	service install	service start
service status	service stop	

Examples:

```
service start;

service stop;

service status;
```

datasource Commands

This section describes **datasource** commands.

check datasource	create env	create envprop
export datasource	import datasource	show datasource
show datasource prop		

Roles and Privileges

This section describes about the privileges and roles assigned for Discovery users. For more information on user creation, see section Create and edit Web Console users and groups in *Structured Data Manager - DiscoveryRuntime Guide*

The following are the lists of Discovery users' roles and privileges.

Privileges	Roles	Description
Administrative	SDM Admin	Indicates the user is the SDM Administrator. Having the Authenticated Administrator privileges does not imply any of the other privileges. Users with this privilege still require the other privileges to perform those functions.
Access Discovery		Can log in to Discovery.
Discovery administrative	Admin	Indicates the user is the Administrator for Discovery. Having the Authenticated Administrator privileges does not imply any of the other privileges. Users with this privilege still require the other privileges to perform those functions.
Discovery management	Manager	Indicates user have Discovery management privileges
Discovery view, review and report	Operator	Indicates user have the access to view, review, and report on Discovery
View data discovery	Data Operator	Indicates user have the privilege to view the data from Discovery.

Discovery user functionality

The following is a list of Discovery user role-based functional accessibility.

Tabs	Sub-Tabs	Roles / Operations	Admin	Manager	Operator	SDM Admin	Data Operator
Dashboard		Create New Project	✓	✓		✓	
Sources		All	✓	✓		✓	
Project		Create new project	✓	✓		✓	
		Edit Projects	✓	✓		✓	
	Dashboard	Edit Project	✓	✓		✓	
		Project scan settings	✓	✓		✓	
		Delete project	✓	✓		✓	
		Scan blackouts	✓	✓		✓	
		Import /Export projects	✓	✓		✓	
		Remediations	✓	✓		✓	
	Schemas	Register Schemas	✓	✓		✓	
	Tables (Table Details)	Register Tables	✓	✓		✓	
		Delete tables and Duplicate tables	✓	✓		✓	
		Scan	✓	✓		✓	
		Reset	✓	✓		✓	
		Deeprest	✓	✓		✓	
		Freeze and Unfreeze	✓	✓		✓	

		Schedule scans	✓	✓		✓	
		Review Tables	✓	✓	✓	✓	
		Tag / Comment	✓	✓	✓	✓	
		Hide / Show details	✓	✓	✓	✓	
	Tables (Column Details)	Actions	✓	✓		✓	
		Review columns	✓	✓	✓	✓	
		View sample of Data				✓	✓
Schedule	Scheduler	Change Status	✓	✓		✓	
		Exec On	✓	✓		✓	
		Delete schedules	✓	✓		✓	
		Start Discovery service	✓	✓		✓	
	Service commands	Run scripts through Discovery service	✓			✓	
	Command Log	View Command logs	✓			✓	
	Service Log	View Service logs	✓			✓	
Grammars	All Tabs	Create /Edit / Delete Class	✓			✓	
		Create /Edit / Delete Sets	✓			✓	
		Create /Edit / Delete Set rules	✓			✓	

			✓			✓	
		Create /Edit / Delete Type	✓			✓	
		Create /Edit / Delete Masking	✓			✓	
		Create /Edit / Delete Groovy	✓			✓	
	Sets	Publish	✓			✓	
Configuration	Scanner	Edit Configurations	✓	✓		✓	
	UI		✓	✓		✓	
	View data		✓	✓		✓	
	Scan blackout		✓	✓		✓	
	All Tabs	Analyze Discovery tables				✓	
		Refresh Discovery MVs				✓	
Logs		Delete Logs	✓			✓	

Troubleshooting

This section describes how to problems symptoms cause and the solutions.

- Solr search on different hostname/port on discovery
- On SQL Server, intrusive environment schema list is not shown on discovery
- Parameter values are not updated when the command is executed from the CLI
- Discovery service unable to start on discovery
- Discovery service unable to schedule jobs for scan

Solr search on different hostname/port on discovery

Symptom

Solr indexes are created based on the parameter `wcurl`, which has the value `http://localhost:8080/WebConsole/ext/Discovery`. Indexes created with this `wcurl` do not work if your Web Console is running on a different host or port.

Resolution

- Update the `wcurl` parameter value before creating the indexes. Run the below command from discovery CLI

  `set param wcurl=http://<hostname>:<port>/WebConsole/ext/Discovery`

- Exit and relaunch CLI to update the parameter value

On SQL Server, intrusive environment schema list is not shown on discovery

Symptom

On SQL server intrusive environment, schemas are unavailable to add to the project.

Cause

The interface user does not have read permission to the database and hence do not have access to the schemas.

Resolution

Provide read permission to the user on the schema to be used

For example, on SQL Server, go to user mapping properties of the interface user > Select the database to be used on discovery > Give db datareader permission for the selected database.

Parameter values are not updated when the command is executed from the CLI

Symptom

Parameter values are not updated when the command is executed from the CLI

Resolution

Since the parameter values are cached, exit CLI and relaunch to reflect the changes values

Discovery service unable to start on discovery

Symptom

Discovery service installed and launched successfully, but on the Discovery, it does not show up under active services.

Cause

- The service when started might neither have run as administrator nor granted all the required permissions

- On the service log we can see the error as `<OBTHOME>\config\obtdiscovery.properties` `(Access is denied)`

Resolution

Grant read, write, and execute permission to the properties file for the user running the discovery service.

Discovery service unable to schedule jobs for scan

Symptom

Discovery service installed and launched successfully and jobs for the scan were scheduled successfully, but the scan does not happen, and the scheduler did not list the registered job.

Cause

The service would have started with a different `srvid` which is not updated under project scan settings before scheduling a job for the scan.

Resolution

- Update the project scan settings for the `srvid` to be used for the scan which is up and running.

- For SDM integrated with Fusion platform in the Cloud:

  ◦ Edit `<OBTHOME>/config/fusionapp.properties` file

  ◦ Update `service.scan.id=srvid` which is used for scan requests from the cloud.

Appendixes

The following are the list of appendixes:

Appendix A: Example of custom grammars

Appendix B: Example of custom created dictionaries

Appendix A: Example of custom grammars

- Birthdate:
  - ```
    <PATTERN name="Birthdate yyyy mm dd"

        regex="\b([12]\d{3}[-\/.]?(0[1-9]|1[0-2])[-\/.]?(0[1-9]|[12]\d|3[01]))\b"

        locale="fr"

        enabled="1"

        img="flags/fr.png" />
    ```
  - ```
    <PATTERN name="Birthdate mm dd yyyy"

        regex="\b((0[1-9]|1[0-2])[-\/.]?(0[1-9]|[12]\d|3[01])[-\/.]?[12]\d{3})\b"

        locale="fr"

        enabled="1"

        img="flags/fr.png" />
    ```
  - ```
    <PATTERN name="Birthdate dd mm yyyy"

        regex="\b((0[1-9]|[12]\d|3[01])[-\/.]?(0[1-9]|1[0-2])[-\/.]?[12]\d{3})\b"

        locale="fr"

        enabled="1"

        img="flags/fr.png" />
    ```
- French National ID:
  - ```
    <PATTERN name="French National Identity Card ID"

        regex="\b\d{12}\b"

        locale="fr"

        enabled="1"

        img="flags/fr.png" />
    ```
- French SSN:
  - ```
    <PATTERN name="French SSN Without Checksum"

        regex="\b[1-478][ ]?\d{2}[ ]?\d{2}[ ]?[0-9][0-9AB][ ]?\d{3}[ ]?\d{3}\b"

        locale="fr"

        enabled="0"

        img="flags/fr.png" />
    ```
  - ```
    <PATTERN name="French SSN With Checksum"
    ```

```
                regex="\b[1-478][ ]?\d{2}[ ]?\d{2}[ ]?[0-9][0-9AB][ ]?\d{3}[ ]?\d{3}[ ]?\d
        {2}\b"

                locale="fr"

                enabled="1"

                img="flags/fr.png"

                groovyfile="validate"

                methodname="isFrenchSSN" />
```

- French Driving License:
  - ◦ <PATTERN name="French License"

        ```
                regex="\b\d{6}[ .-]?\d{6}\b"

                locale="fr"

                enabled="1"

                img="flags/fr.png" />
        ```

- French Passport:
  - ◦ <PATTERN name="French Passport"

        ```
                regex="\b\d{2}[A-Za-z]{2}\d{5}\b"

                locale="fr"

                enabled="1"

                img="flags/fr.png" />
        ```

- French Phone Number:
  - ◦ <PATTERN name="French Formatted"

        ```
                regex="\b((\+|00)33[\s.-]{0,3}(\(0\)[\s.-]{0,3})?|0([\s.-]{0,3})?)[1-9](([\s.-
        77]?\d{2}){4}|\d{2}([\s.-]?\d{3}){2})\b"

                locale="fr"

                enabled="1"

                img="flags/fr.png" />
        ```

- French Postal Code:
  - ◦ <DATAFILE name="French Postal Code"

        ```
                file="postalcodes.fr.txt"

                locale="fr"

                enabled="1"

                img="flags/fr.png" />
        ```

Appendix B: Example of custom created dictionaries

French postal codes dictionary file:

```
24000
24001
24002
24003
24004
24005
24007
24009
24010
24011
24012
24013
24014
24015
```

Updated `Validate.groovy` file containing the validation function for the frenchSSN:

```groovy
boolean isFrenchSSN(String input) {
    if (ssn.size() != 15) {
        return false
    }
    def matcher = (ssn =~ /(([1278])(\d{2})(\d{2})(\d{5}|2[abAB]\d{3})(\d{3}))(\d{2})/)
    if (!matcher.matches()) {
        return false
    }
    long id = Long.parseLong(ssn.substring(0, 13).replaceAll(/2[aA]/, '19').replaceAll(/2[bB]/, '18'))
    long controlKey = Long.parseLong(ssn.substring(13, 15))
    long calculatedControlKey = 97L - (id % 97L)
    return (calculatedControlKey == controlKey)
}
```

`frenchSSN.groovy` file that only contains the function to validate the frenchSSN. This can be used with 'Groovy' console to validate the function.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on OpenText Structured Data Manager - Discovery 24.3.100 SDM - Discovery Guide

Add your feedback to the email and click **Send**.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to swpdl.sdm.docfeedback@microfocus.com.

We appreciate your feedback!