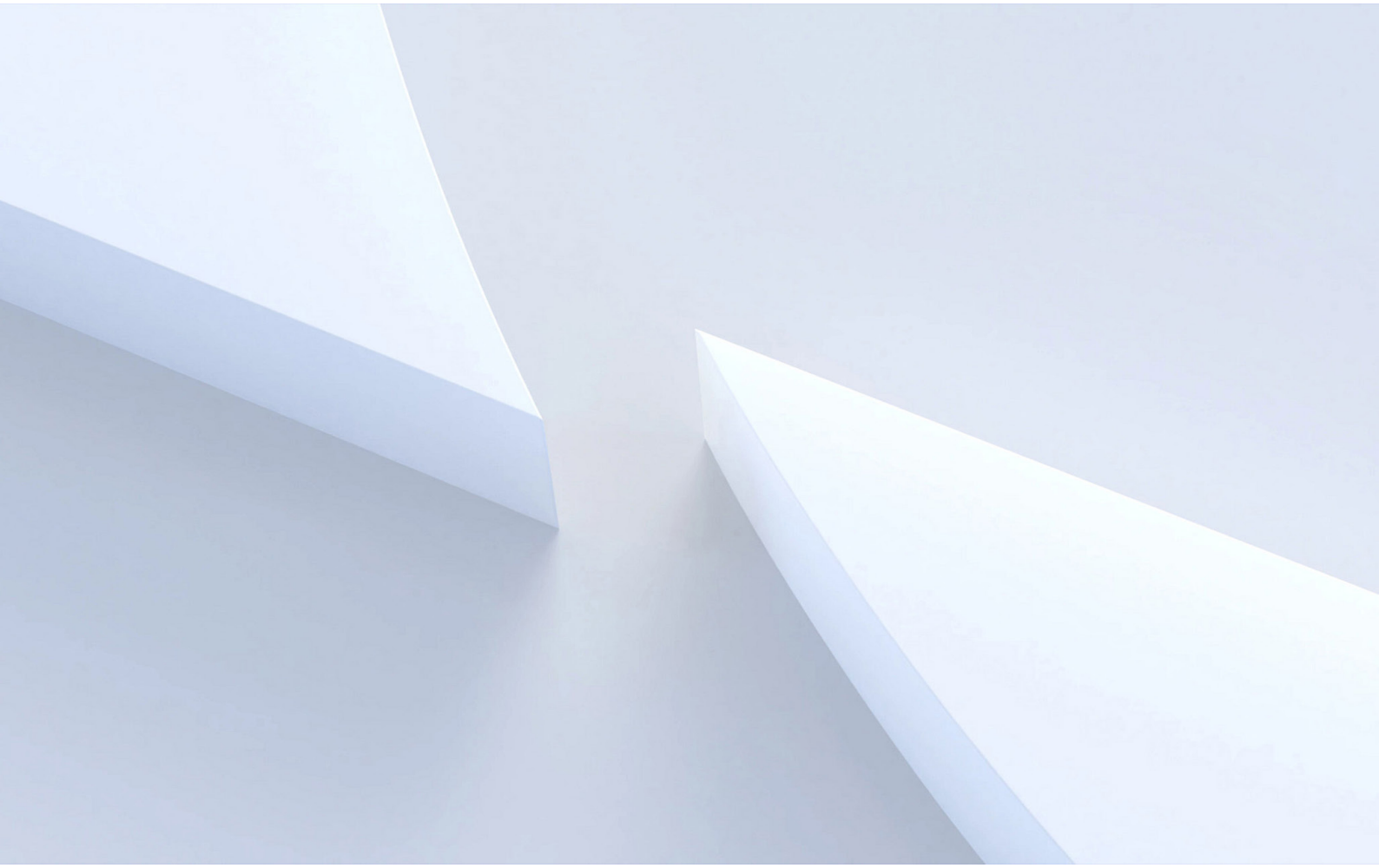


ProSource 2023.2

Minor Release Installation Guide

Contents

Copyright 3

Introduction 4

ProSource Installation 5
- Prerequisites 5
- Pre-installation Task 5
- Installation Steps 6

Post Installation Tasks 8
- Update and Restore Old Pages and Collection 8
- Secure RMI with HTTPS Configuration 8
- Overview of Microservices 11
- Software Installations for Document Viewers/Scanners 13
- Deploy Microservices on the ProSource Server (Locally) 13
- Data Ingestion to OSDU 27
- DataSource Caching for Faster Performance in Pages and Collection 27
- Configuration for delta updates in pages 28
- Uploading Large Files 28
- Mandatory attributes for logs loading 28
- Document Scanner Configuration 29
- Document Viewers 31
- File Upload 31
- Search within the data file configuration 32
- Set Templates Priority for the Seismic Auto- Scanner 33
- Deploy Microservices on a Remote Server 34

ProSource additional configuration and procedures 37
- Configuration of SSL LDAP for ProSource 37
- Change SIS_ADMIN Password 37
- Log in to Oracle account using DB token 37
- Gunicorn configuration to improve performance on dashboard pages 38
- Configuration for Plugin transfer 39

Troubleshooting 40
- ProSource Installer reports zero space available for disks with 1+ TB free space 40
- ProSource upgrade installer fails giving OpenLDAP error 40
- Recover from a failed/cancelled/aborted ProSource installation 40
- Folders/dropbox directories do not display scanner/loader file explorer due to fewer threads 41
- Microservices Related Common Issues 41
- Job Definitions Restoring 41
- Seabed Extension export failure 41
- Seabed utility package CALC_WELL_POINT is not working 42
- Interface creation on new project 42

Copyright

Trademarks and Service Marks

The word "Schlumberger", the word "SLB", the SLB logo, and other words or symbols used to identify the products and services described herein are either trademarks, trade names or service marks of SLB and its licensors, or are the property of their respective owners. These marks may not be copied, imitated or used, in whole or in part, without the express prior written permission of SLB. In addition, covers, page headers, custom graphics, icons, and other design elements may be service marks, trademarks, and/or trade dress of SLB, and may not be copied, imitated, or used, in whole or in part, without the express prior written permission of SLB. Other company, product, and service names are the properties of their respective owners.

ProSource® is a mark of SLB.

An asterisk (*) is used throughout this document to designate other marks of SLB.

Security Notice

The software described herein is configured to operate with at least the minimum specifications set out by SLB. You are advised that such minimum specifications are merely recommendations and not intended to be limiting to configurations that may be used to operate the software. Similarly, you are advised that the software should be operated in a secure environment whether such software is operated across a network, on a single system and/or on a plurality of systems. It is up to you to configure and maintain your networks and/or system(s) in a secure manner. If you have further questions as to recommendations regarding recommended specifications or security, please feel free to contact your local SLB representative.

Introduction

ProSource 2023.2 is a minor release that provides updates to the ProSource 2023.1.

This version supports the Red Hat Enterprise Linux Server release 7.9 version and SLB Licensing Tool 2021.1. It is compatible with ArcGIS Server 10.9.1 and JRE 1.8 Update 181 (32-bit) or OpenJDK 8 (32-bit) along with OpenWebStart for client.

It is compatible with the following components:

- An existing ProSource 2023.1 or ProSource 2023.1 BCI 01.
- ArcGIS 10.9.1 and JRE 1.8 Update 181 (32-bit) for client.
- Google Chrome 114 (For ProSource Web only).
- The recommended screen resolution for best user experience is 1920x1080.

ProSource Installation

This section contains instructions on how to install ProSource 2023.2.

Prerequisites

Before you upgrade the ProSource, confirm that the following prerequisites are met:

- ProSource 2023.1 or ProSource 2023.1 BCI 01 is installed.
- All ProSource licenses are valid.
- The ProSource database and listener are up and running.
- The user management service used for ProSource (OpenLDAP or Active Directory) is running and accessible from the ProSource server.
- All necessary environment variables are set in order to run SQL*Plus (i.e. PATH, LD_LIBRARY_PATH, ORACLE_HOME, ORACLE_SID).
- Make sure PS_HOME is defined in .bash_profile or .bashrc file. If not then set it and then reopen the linux terminal before proceeding further.

Pre-installation Task

ProSource allows some customization such as https protocol and port changes.

Back up the files mentioned below and refer to same post-installation steps to speed up reconfiguration.

`$PS_HOME/conf/Tahiti/ps_rmi.properties`

`$PS_HOME/conf/prosource.xml`

`$PS_HOME/TomcatHome/conf/catalina.properties`

`$PS_HOME/TomcatHome/conf/server.xml`

`$PS_HOME/TomcatHome/webapps/imadmin/template/imadmin.jnlp.template.txt`

1. During installation, ProSource deploys the new job_definitions file. Administrators can use the backup file to reconfigure the customized job polling directory and job queue settings. To back up the job definitions file, go to `$PS_HOME/conf/tahiti`.

   Copy the `job_definitions.xml file` and rename the new file as `job_definitions_backup.xml`.
2. To backup the ProSource Web Dashboard templates, Charts, View Data pages, and collections, back up the below files and refer same post-installation steps to speed up the reconfiguration.

   `$PS_HOME/microservices/prosource-dashboard/dash_directory/Layouts`

   `$PS_HOME/microservices/prosource-dashboard/dash_directory/Collections`

   `$PS_HOME/microservices/prosource-dashboard/dash_directory/Datasources`

   `$PS_HOME/microservices/prosource-dashboard/dash_directory/CacheConfig/config.json`

   `$PS_HOME/microservices/prosource-dashboard/dash_directory/templates`

`$PS_HOME/microservices/prosource-dashboard/dash_directory/SavedConfigFiles`

`$PS_HOME/microservices/prosource-dashboard/config/graphList.xml`

3. Back up the following configuration files:

- `$PS_HOME/microservices/sourceMS.config`
- `$PS_HOME/ microservices/scanner_standalone/dropbox.config`
- `$PS_HOME/microservices/prosource-configuration-services/config/logsloading.configuration.json`
- `$PS_HOME/microservices/prosource-configuration-services/config/contractor.config.json`

Note: If possible, take a restore point or backup of the ProSource and oracle server system to revert in case the installation fails or uninstallation is required.

Note: The ProSource minor release installer backs up the files that are overwritten during installation. The backup is available at `$PS_HOME/ uninstall`. There is a known issue with uninstallation, so if possible, take a restore point or backup of the ProSource and oracle systems.
During installation, approximately 15 GB of disk space is required to create and compress the backup but it can be more depending on the cache of datasources. After installation, approximately 5 GB disk space is required. If you remove the backup file you cannot restore it later. In general, we recommend you to have 45 GB of free space including temp before the installation.

The Linux machine tmp directory or Linux user home directory should also have a minimum of 15 GB free space.

Note: The ProSource service pack alone is of 5.4 GB size, so it is recommended that you maintain at least 35 GB of space in your installer environment for proper unzipping and installation of the ProSource service pack.

Warning: It is recommended not to cancel the installation process until it is complete.

Installation Steps

To install the ProSource 2023.2 minor release, complete the following steps:

1. Stop the ProSource and Tomcat servers using the following commands respectively:

`$PS_HOME/bin/prosource-server.sh stop`

`$PS_HOME/TomcatHome/bin/shutdown.sh`

Note: When you are upgrading ProSource, make sure you stop all the microservices using the command: `./stopMS.sh` and then choose option a. To ensure all the microservices are stopped correctly, run the command: `./startMS.sh -s`.

2. Copy the ProSource Service installer zip file to an empty temporary directory. The Service installer kit file is named as follows:

`PS2023.2_C<build_number>_<timestamp>_<branch_name>_Linux. zip,`

where <build> is the build number.

3. Unzip the contents of the kit using the following command:

   unzip `PS2023.2_C<build_number>_<timestamp>_<branch_name>_Linux. zip`

   This command extracts the actual installer executable and associated files.
4. Launch the installer executable:

   `./PS2023.2_C<build_number>_<timestamp>_<branch_name>_Linux. bin`

   where <build> is the build number.

   The **Introduction** dialog box appears.
5. Review the installation prerequisites and instructions, and then select **Next**

   The **Oracle Home** dialog box appears.
6. Enter the **Oracle Home Directory Location** and the **TNS_ADMIN Directory Location** of the Oracle instance where Seabed is installed.
7. Select **Yes** or **No** to indicate whether you want the installer to determine the Oracle instance, and then select Next.
   - If you select Yes, the Seabed Instances dialog box appears.
   - If you select No, you must provide this information manually.

Note: If the **SIS_ADMIN** account is locked, the installer does not detect any Seabed instances and the installation process stops. To proceed, unlock the **SIS_ADMIN** account and then rerun the installer.

8. Enter the connection parameters: **Oracle Host**, **Oracle Port**, **Oracle SID**, **SIS_ADMIN** password, and then select **Next**.
   The Pre-Installation Summary dialog box appears.

Note: If you incorrectly enter the **SIS_ADMIN** password multiple times, and your attempts exceed the Oracle password policy, the **SIS_ADMIN** account locks. To complete the installation, unlock the **SIS_ADMIN** account and then rerun the installer.

9. Verify the details, and then select **Install**.

   The installer displays the **Install Complete** window.
10. Select **Done**.

Note: Refer to the log files to ensure that there are no errors during installation. The log files are located in the following directory: `$PS_HOME/install`

11. Start the ProSource and Tomcat servers using the following commands:

    `$PS_HOME/bin/prosource-server.sh start`

    `$PS_HOME/TomcatHome/bin/startup.sh`
12. Complete the post installation tasks outlined in the section below.

Post Installation Tasks

This section provides post-installation instructions for ProSource.

Update and Restore Old Pages and Collection

This is an optional step.

This step is required if the admin has created or modified pages or collections in the earlier ProSource version. To merge and update the modified pages and collection from a previous ProSource installation to the current ProSource installation, follow the steps below:

1. Copy all the files from the pre-installation backup folders for below locations and replace those in the corresponding post installation folders:
   - `PS_HOME/microservices/prosource-dashboard/dash_directory/Datasources`
   - `PS_HOME/microservices/prosource-dashboard/dash_directory/templates`
   - `PS_HOME/microservices/prosource-dashboard/dash_directory/savedConfigFiles`
2. Go to `$PS_HOME/microservices/prosource-dashboard/dash_directory` and copy the folder **Layouts** by name `Layouts_current_version` then clear the content inside **Layouts** folder.
3. Similarly go to folder `$PS_HOME/microservices/prosource-dashboard/dash_directory` and take backup of folder **Collection** as `Collections_current_version` then clear the content inside **Collections** folder.
4. Go to `$PS_HOME/microservices/prosource-dashboard` and run the script `upgrade_collections_and_pages.py` in the following order.
   For merging pages:

   ```
   $PS_HOME/microservices/python-modules/dashboard_env/bin/python3.9
   upgrade_collections_and_pages.py -t layout -e
   <Folder_Path_for_previous version_Layouts_backup> -n
   <Folder_Path_for_current version_Layouts_backup> -o $PS_HOME/
   microservices/prosource-dashboard/ dash_directory/Layouts
   ```

   For merging collections:

   ```
   $PS_HOME/microservices/python-modules/dashboard_env/bin/python3.9
   upgrade_collections_and_pages.py -t collection
   -e <Folder_Path_for_previous version_Collections_backup> - n
   <Folder_Path_for_current_version_Collections_backup> -o $PS_HOME/
   microservices/prosource-dashboard/dash_directory/Collections
   ```

Secure RMI with HTTPS Configuration

If Secure RMI with HTTPS is already configured from last installations then please follow only Step 6 and 7, otherwise follow all the steps outlined below.

1. Create a Certificate keystore file.

For configuring secure RMI and secure HTTPS, create a certificate keystore file first. Follow these steps to create a certificate:
Generate keys and keystore file at `<PS_HOME> /resources/ps_cert.jks`:

```
$ mkdir <PS_HOME>/resources

$ cd <PS_HOME>/resources/

$ <PS_HOME>/jre64/bin/keytool -genkey -alias <ALIAS_NAME> - keyalg RSA
-keystore ps_cert.jks -storepass <STORE_PASSWORD>
```

2. Create a soft link to the keystore file from ProSource Web by typing the following command:
   `ln -s <PS_HOME>/resources/ps_cert.jks <PS_HOME>/TomcatHome/ webapps/psweb/`
3. Run the IMAdmin Console in safe-start mode and modify the TOMCAT_URL property:
   Launch the IMAdmin console with the following command:
   a) `cd <PS_HOME>/TomcatHome/webapps/imadmin/`
   b) `./safe-start-imadmin.sh`
   c) Login with sis_admin and click the **ProSource** tab.
   d) Double-click the FederatorConfigurationManagement component and set the appropriate Tomcat URL under the Tomcat Server URL as follows.
   Error!Hyperlink reference not valid.
   e) Click **OK** and then close the application.
4. Configure Secure RMI
   Follow these steps to configure secure RMI server:
   a) To start the ProSource server in secure RMI mode, create a new file `<PS_HOME>/conf/tahiti/ps_rmi.properties` with the following text:
   # To start the ProSource server in secure RMI mode jnlp.ps_secured=true
   jnlp.storeLocation=<PS_HOME>/resources/ps_cert.jks
   storePwd=<STORE_PASSWORD>
   alias=<ALIAS_NAME>

   Note: As the file ps_rmi.properties contains critical information, ensure it is safeguarded by the local IT department.

   b) Edit the file `<PS_HOME>/TomcatHome/conf/catalina.properties`, and modify the `shared.loader` property into:
   c) `shared.loader="<PS_HOME>/conf/tahiti/"`
5. Configure Secure HTTPS
   a) To enable the Tomcat HTTPS server (port 8443), edit the file `<PS_HOME>/ TomcatHome/conf/server.xml`:
   b) Delete or comment the following command:
   `<Connector port="8090" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />`
   c) Uncomment the following command:

```
<Connector port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol" maxThreads="150"
SSLEnabled="true" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
```

And replace with the following command:

```
<Connector port="8090" protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" server="ProSource"/>

<Connectorport="8443"

protocol="org.apache.coyote.http11.Http11NioProtocol" maxThreads="150"
SSLEnabled="true" scheme="https" secure="true"

keystoreFile="<PS_HOME>/resources/ps_cert.jks"

keystorePass="<password>" clientAuth="false" sslProtocol="TLS" />
```

Note: keystorePass must be the password provided during step#1 of certificate creation.

Note: As the file server.xml contains critical information, ensure it is safeguarded by the local IT department.

6. To set properties for the ProSource client, edit the file path `<PS_HOME>/conf/prosource.xml` and add the following command after this line: `<!-- PROSOURCE_SEISMIC_END -->`

```
<property name="jnlp.ps_secured" value="true" />

<property name="jnlp.storeLocation" value="http://
PROSOURCE.HttpServer:8090/psweb/ps_cert.jks" />
```

7. To set properties for the IMAdmin client, edit the file path `<PS_HOME>/ TomcatHome/webapps/ imadmin/template/ imadmin.jnlp.template.txt` and add the following command after this line: `<argument>IMADMIN.UserData</ argument>`:

```
<argument>-jnlp.ps_secured</argument>

<argument>true</argument>

<argument>-jnlp.storeLocation</argument>

<argument>http://IMADMIN.HttpServer:8090/psweb/ ps_cert.jks</argument>
```

8. Configure the following properties in sourceMS.config::

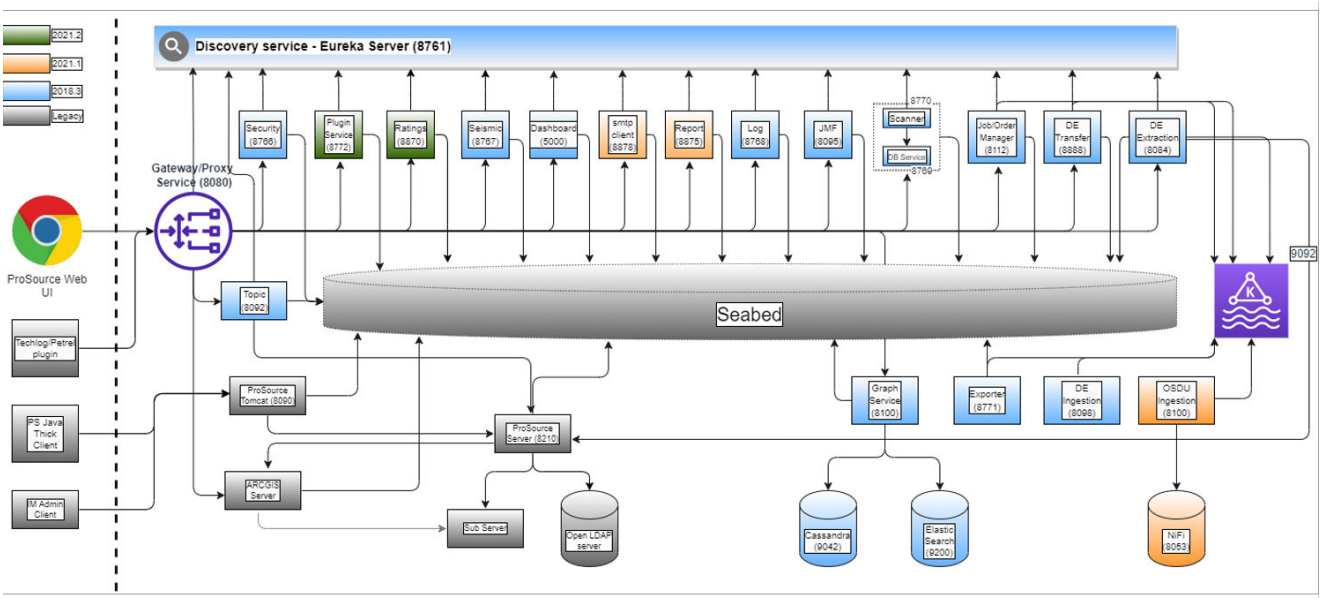
```
export JNLP_PS_SECURED=true

export JNLP_STORE_LOCATION=<PS_HOME>/resources/ps_cert.jks

export STORE_PASSWORD=<password>

export ALIAS=<alias>
```

Note: Password and alias should be same as provided in Step 1 of certificate creation.

9. Restart the ProSource and Tomcat server.
10. Launch the IMAdmin page through URL as below:
    https://server_ip_or_hostanme:8443/imadmin/webstart

Overview of Microservices

This section gives the brief overview of various microservices.



Gateway Service: It is a gateway for all the incoming client requests. It will intercept all the requests and then route the request to the actual services. It uses discovery service for getting actual service details (host, port).

The Gateway service will run at the 8080 port and this port need to be opened on the firewall.

Discovery Service: Discovery service registry is a phone book for your microservices. Each service registers itself with the service registry and tells the registry where it lives (host and port). Discovery enables services to find and communicate with each other without hard-coding hostname and port in each service.

Default port: 8761

Security Service: This service is responsible for authenticating a user. It supports both LDAP and Active Directory. All requests are authenticated and goes to the actual service on a valid credential request.

Default port: 8766

Job Manager: Job Manager is one of the microservice which is responsible for accepting seismic and well data loading, exporters and manual scan job requests. It schedules the job and run it as per the schedule configuration.

Seismic Service: The Seismic service is responsible for getting all seismic related information such as Data-store, Seismic projects, surveys etc from database. Also, Histogram, Trace Viewer uses seismic API's to read data from seismic files.

Log Service: The Logs service is responsible for displaying well related metadata such as UBHI, CRS, Contractor, Date format etc used while data loading workflow. This service also drops the PSDC file in the desired polling directory.

Scanner Service: The Scanner service is responsible for pre or manual scanning the LAS, DLIS, LIS, SEGY, SEGD, ZGY, digital text documents (PDF, TIF, and TIFF), Navigation, and RODE files to get the information about the file and saves this information in the JSON file format.

DB Service: The DB service is responsible for saving the JSON file in the Oracle database which gets generated by Scanner.

Extractor Service: There are some default templates provided along with the kit. These templates are representation of the tree structure topics present in ProSource. Therefore, it is a micro-service which is used to extract the data for the topics present in the template which has been selected from the UI.

Exporter Service: The exporter service is responsible for exporting the data files from the ProSource database to the user's local environment or another server.

Dashboard Service: The Dashboard service enables users to visualize data stored in the ProSource database. Using the Dashboard, you can create new charts and custom dashboard templates.

Topic Service: The topic service has APIs for the CRUD operation on ProSource topics. Data in ProSource Web are read/manipulated through different APIs defined in this service. Make sure that whenever you restart this service you need to re-login the ProSource Web for any CRUD operation on data.

Ingestor Service: The ingestor service is created for ingesting ProSource data into DELFI. There are no APIs exposed in this service rather this service communicates with Kafka to get the data/ messages to ingest. This service has defined the consumer, which reads the data from topics defined in kafka and ingest into DELFI.

Transfer Service: This service act as an intermediary between ProSource Web and Extractor Service while ingesting the data into DELFI. This service communicates with Kafka. This Service has defined the producer to send the message on Kafka's topic. When a user exports data to DELFI from ProSource Web, this service builds the message and send it to Kafka which in turn is consumed by a consumer defined in the extractor service.

Email service: This service enables ProSource to send email notifications to intended recipients using configured mail server for the Export workflow.

Report service: With this service ProSource user is allowed to configure & generate the BIRT (Business Intelligence and Reporting Tools) and SSRS (SQL Server Reporting Service) reports. Configured reports fetch the data from database and displayed in the designed report format. The generated reports can be downloaded and saved.

OSDU Ingestor Service: The OSDU ingestor service is created for ingesting ProSource data into OSDU. It is possible to ingest Seismic, logs, and CSV data from various data topics.

OSDU Ingestor Token Service: This OSDU service generates the bearer token for OSDU. This service defines one rest API which generates the bearer token for OSDU. The rest service defined here takes scope and secret as inputs. Both scope and secret are encrypted.

Plugin Service: The Plugin service is created to support Techlog and Petrel plugins to transfer data from ProSource to Techlog/Petrel. It has APIs which are used by plugin to fetch data from seabed.

Rating Service: The 5-star rating system allows users to rank their feedback on a 5-point scale from 1 to 5. Users can provide their 5-star ratings to the selected records for a particular topic. Based on the ratings provided by multiple users, Rating service calculates the overall rating and store along with their comments.

Graph Controller Service: This service is used to build the smart cache with the capabilities of various types of indexing mechanisms. Based on which the large amount of data is fetched from this cache efficiently. Also, it takes the delta updates from the source database at the regular interval based on the scheduler time which is configurable.

Software Installations for Document Viewers/Scanners

You can view and scan documents of various extensions in ProSource. You can skip these steps if the document viewer/scanner installation was done in a previous release installation.

Download the software `doc-scanner-packages.zip` that are packaged along with the ProSource 2023.1 release collection from the SLB SDC site and copy it to the ProSource Server environment where the document scanner is supposed to be working. For example, copy the software package to the `/home/install/ Downloads` location.

1. Unzip the software package.
2. Run the command tar `-xzvf tesseract-5.0_v2.tar.gz` from the path where the software is downloaded.
3. Run the command tar `-xzvf leptonica-1.79.0.tar.gz` from the path where the software is downloaded.
4. MAGICK_HOME must reside under `/usr/local only`. Log in via root user into the ProSource server and grant install user permission to access `/usr/ local` folder using the following command.

   ```
   chown -R install:install /usr/local
   ```

   Keep `ImageMagick-7.1.0-19.tar.gz` under `/usr/local` and run the command `tar -xvzf ImageMagick-7.1.0-19.tar.gz` from this path.
5. Download ProSource 2023.1 - LibreOffice for Office Documents Viewer from SDC and copy it to `/home/install/Downloads`
6. Log on via root user into the ProSource server and run the following command:

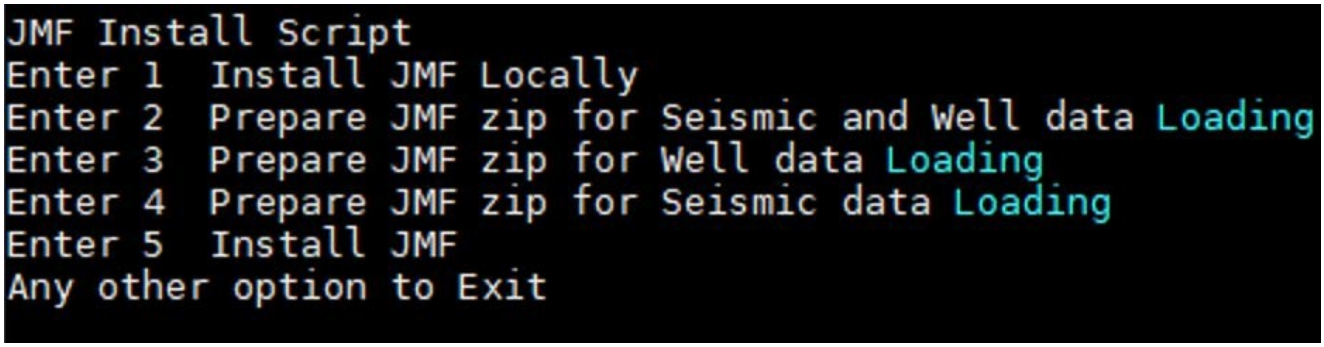
   ```
   tar -xvf LibreOffice_6.4.4_Linux_x86-64_rpm.tar.gz
   cd LibreOffice_6.4.4.2_Linux_x86-64_rpm/RPMS/
   yum install *.rpm
   ```
7. Navigate to opt directory using the command:

   ```
   cd /opt
   ```
8. Create a softlink to libreoffice home using the command below:

   ```
   ln -s /opt/libreoffice6.4 /usr/lib/libreoffice
   ```
9. Log on via install user into the ProSource server, then navigate to Microservices home and restart the scanner service.

Deploy Microservices on the ProSource Server (Locally)

1. Go to the path `$PS_HOME/microservices/prosource-jmf`.
2. Source or export environment variables `PS_HOME and ORACLE_HOME`
3. Run the `./jmf_make.sh` command.



a) To install Job Management Framework (JMF) on the same server where ProSource is installed, enter option 1 only.

b) Enter the existing ProSource Home Path, for example `$PS_HOME`.

c) Enter a location path where you want to install Job Management Framework. A new folder is created by script automatically if a folder is not present.

Note: The JMF folder must be set outside ProSource Home.

It is recommended not to delete the JMF installation directory. To install JMF at a new location, the previously installed JMF can be uninstalled using the `Uninstall_JMF` script present under JMF_HOME (the location where JMF has been installed).

4. Follow the below steps to uninstall JMF:

   a) Go to **JMF HOME**.

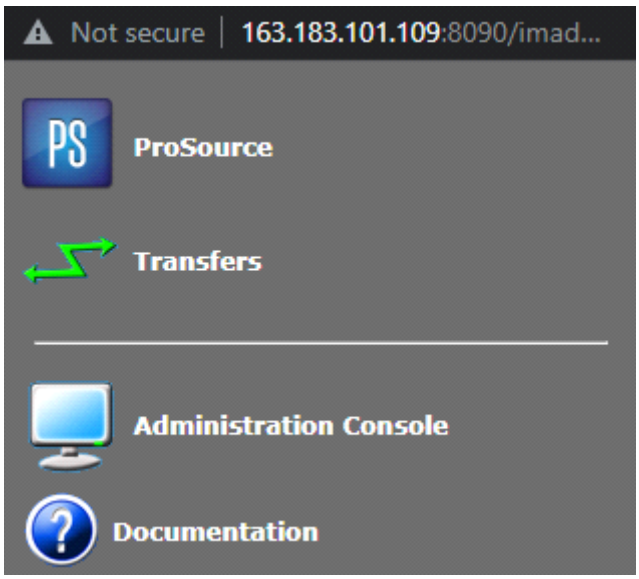
   b) Run the script `Uninstall_JMF.sh` and provide the `JMF_Home` path

Note: Give the execute permission to the script `Uninstall_JMF.sh`. If it does not have, use a command `chmod a+x Uninstall_JMF.sh`.

5. Go to the path `$PS_HOME/microservices/ui`.
6. Unzip the `XXXXX_dist.zip` by using the command `unzip -a XXXXX_dist.zip`
7. Go to the `$PS_HOME/microservices/ui/dist/assets` path, and then open the `settings.json` file and make the changes to the attributes as per the table below.

Attribute name	Action
"proxy"."host"	Set this to the hostname where microservices are deployed (if microservices are deployed on the same machine as the ProSource server, then this hostname will be same as the ProSource server hostname.
"proxy"."port"	Set the port for the web application. The default port is 8080. **Note:** Make sure that the proxy port is opened on the firewall to access the ProSource web.
"proxy"."protocol"	Set it to either "http" or "https".

Attribute name	Action
"connectedtoDELFI"	This is for the visibilty of the exporttoDelfi button on the UI. When a property is set to true, button is available however when a property is set to false then the button does not appear on the UI.
documentDownload LimitInMB	This property defines the download limit for file size from the ProSource Web modules.
mapRootService	Set this to the ArcGIS service folder where you have published services for Map View.
downloadZipFileLimitInGB	Size of zip download which will be allowed from ProSource Web.
smartCacheEnabled	This is for the availability of the Smart Cache (Beta) feature on the UI. When a property is set to *true*, Smart cache feature is available however when a property is set to *false* then the Smart cache feature is not available.
enableExternalApps	The enableExternalApps flag enables integration of external apps with ProSource Web. With this integration, users can launch and access in-house/ external applications directly from the ProSource interface.
sauthSSOEnabled	This property should be set to *true* only if SSO needs to be enabled via DELFI ecosystem.
azureSSOEnabled	This property should be set to *true* only if SSO needs to be enabled via Azure Active directory.
IsActiveDirectoryConfigured	Set this flag to true if for enabling Register User workflow for Active Directory configured Server.



Note: Ports for proxy service is mentioned in the settings.json file must be opened on the firewall to access the ProSource Web.

8. Configure Dropbox file

Go to the `$PS_HOME/microservices/scanner_standalone/config` path, and then open the `dropbox.config` file.

Note: Before editing the dropbox.config file, ensure that you back up the file and rename it dropbox_backup.config.

a) Configure Scanner/Loader folders:

Set the directory tag with input and output directories. The Scanner crawls and scans metadata for all data files including SEG-Y, LAS, and DLIS present in the input directory and sub-directories. The Scanner dumps the scanned output in JSON format in the output directory by creating the same directory structure as of the input directory. The flag autoScan is set to true if the files in the input directory must be automatically scanned as soon as the scanner service starts. It can be set to false if the user does not want to automatically scan the files and is interested in submitting the manual scan jobs for the files in the input directory.

```
<dropbox>
    <directories>
<directory autoScan="true" input="/home/install/auto_scan_input"
output="/home/install/auto_scan_output"/>
<directory autoScan="false" input="/home/install/manual_scan_input"
output="/home/install/manual_scan_output"/>
    </directories>
```

Note: An admin can configure the system to scan more directories. To do that, add the directory tag following the format shown above.

b) Scanner/Loader dropbox Entitlement

Dropbox directories can be entitled to a project.

When the directory is entitled to a specific project then that directory and its files will be displayed in File Explorer only when user connects to that Project.

Global directories are the directories, which are not entitled. Global directories will be displayed for all projects.

- To make the directories entitled to specific project, please make below changes in dropbox.config.
- New parameter 'seabedProject' is added in the directory configuration.

Blank or skipped seabedProject represents a global directory. Adding a project name entitles specific directory to that project only.

An example of a Dropbox Entitled directory is given below.

```
<directory autoScan="true" input="/home/install/auto_scan_input"
output="/home/install/auto_scan_output/" seabedProject="Project_1"/>
```

Example for Global directory.

```
<directory autoScan="true" input="/home/install/auto_scan_input"
output="/home/install/auto_scan_output"/>
```

Dropbox directories can also be entitled to specific users.

An example of a dropbox entitled directory is:

```
<directory autoScan="true" input="/home/install/auto_scan_input"
output="/home/install/auto_scan_output" entitledUser="user_A",
"user_B"/>
```

c) Enable Scanner formats:

Uncomment scanner tags for scanning files of the respective format. For example, if an admin wants to scan LAS files only, then they can uncomment LAS Scanner only, as shown below

```
        </scanner>
        <scanner
class="com.slb.sis.seisstream.job.dropbox.pollers.datafile.LisDLisScan
ner" name="LIS/DLIS-JSON">
            <extensions>
                <extension value="dlis"/>
                <extension value="lis"/>
            </extensions>
            <properties>
                <entry>
                    <key>PARALLEL_EXECUTION_COUNT</key>
                    <value>4</value>
                </entry>
                <entry>
                    <key>MAX_TIMEOUT_MINUTE</key>
                    <value>1</value>
                </entry>
            </properties>
        </scanner>
```

You can add custom extensions for scanners too as shown below.

```
     </scanner>
        <scanner
class="com.slb.sis.seisstream.job.dropbox.pollers.datafile.SEGDScanner
" name="RODE-JSON">
            <extensions>
                <extension value="rode"/>
    <extension value="rode_1"/>
            </extensions>
            <properties>
                <entry>
                    <key>number-of-records</key>
                    <value>10</value>
                </entry>
            </properties>
        </scanner>
```

Note: More scanners are enabled to scan other types of data files later. The scanner standalone service must be restarted if the scanner configuration is changed.

d) Check and modify PYTHON and PYSCRIPT paths according to server environment as shown in the image below.

e) Provide the template directory path as per the server environment and define the template priority.

Note: Make sure that templates in the file system and database are always in sync.

```
<scanner
class="com.slb.sis.seisstream.job.dropbox.pollers.datafile.SEGYTemplat
```

```
eScanner" name="SEGY-SCN">
            <extensions>
                <extension value="segy"/>
                <extension value="sgy"/>
            </extensions>
            <properties>
                <entry>
                    <key>PYTHON</key>
                    <value>/apps/sis/PS_2023/microservices/python-
modules/auto_scanner_v9_env/bin/python3.9</value>
                </entry>
                <entry>
                    <key>PYSCRIPT</key>
                    <value>/apps/sis/PS_2023/microservices/
scanner_standalone/auto_scanner_v_9/exec.py</value>
                </entry>
                <entry>
                    <key>TEMPLATE_ARCHIVE</key>
                    <value>/CMZDATA/ps_dropbox_2/
auto_scanner_templates</value>
                </entry>
                <entry>
                    <key>TEMPLATE_PRIORITY</key>

<value>DEFAULT_TRACE.sgyfmt,SEGY_REV_1.sgyfmt,SEGY_REV_0.sgyfmt,SEGY_R
EV_1_SX181_SY185_CDPX73_CDPY77.sgyfmt,SEGY_REV_1_IL9_XL13_SP21_SX73_SY
77_CDP17_CDPX183_CDPY187.sgyfmt,SEGY_REV_1_IL5_XL21.sgyfmt</value>
                </entry>
            </properties>
        </scanner>
        <scanner
class="com.slb.sis.seisstream.job.dropbox.pollers.datafile.SEGYFullSca
nner" name="FULL-SEGY-SCN">
            <properties>
                <entry>
                    <key>PYTHON</key>
                    <value>/apps/sis/PS_2023/microservices/python-
modules/segy_scanner_v7_env/bin/python3.9</value>
                </entry>
                <entry>
                    <key>PYSCRIPT</key>
                    <value>/apps/sis/PS_2023/microservices/
scanner_standalone/segy_scanner_v7/segy_scanner_v7.py</value>
                </entry>
                <entry>
                    <key>TEMPLATE_ARCHIVE</key>
                    <value>/CMZDATA/ps_dropbox_2/
auto_scanner_templates</value>
                </entry>
            </properties>
        </scanner>
        <scanner
class="com.slb.sis.seisstream.job.dropbox.pollers.datafile.ZGYScanner"
 name="ZGY-JSON">
            <extensions>
                <extension value="zgy"/>
            </extensions>
        </scanner>
```

f) Refer to the *Document Scanner Configuration* section to configure the document scanners in the dropbox.config file.

9. Configure the Kafka server properties.

These configurations specify the time in milliseconds about minimum and maximum sessions timeout for kafka consumer. This is a mandatory step for all the workflows which include Extractor service.

a) Go to the `$PS_HOME/microservices/kafka/config/` path and open the server.properties file.

b) Add the below variables towards the end of the file.

group.min.session.timeout.ms=1000

group.max.session.timeout.ms=3600000

Note: *session.timeout.ms* in the consumer properties cannot be greater than this group.max.session.timeout.ms

Note: *session.timeout.ms* in the consumer properties cannot be less than this group.min.session.timeout.ms

10. Configure SourceMS.config

Go to `$PS_HOME/microservices/` and open the `sourceMS.config` file.

Note: Back up the `sourceMS.config` file and rename it as `sourceMS_backup.config`.

In this file all the mandatory options must be configured as follows:

Action	Description
Set MS_HOST	MS_HOST refers to the server hostname where the microservices are deployed.
Set DB_HOST	DB_HOST is the server hostname where the Oracle database is deployed.
Set DB_PORT	DB_PORT is the database port where oracle instance is running.
Export PS_HOME	`PS_HOME` is the ProSource Home path. Set path for `PS_HOME` .This is a mandatory property.
Export SEABED_URL	The JDBC connection link to the Oracle database as mentioned in the commented section of the `sourceMS.config` file.
Export ORACLE_HOME	The Oracle home directory where microservices are deployed.
Export ACTIVEMQ_HOME	ActiveMQ is the java-based messaging server. It is installed as part of the ProSource installation.
Export PROXY_PORT	Proxy port must be same as mentioned in `settings.json` file.

Action	Description
Export CONFIG_PATH	This is the path for other configuration files used by PS web. If no value is provided then the default value is: `$PS_HOME/microservices/prosource-configuration-services/config`.
Export ENABLE_SSL	Determines wether the services are accessed via https or the http protocol. Set the value to *False* if the protocol is http and *True* if the protocol is https.
Export KEY_PATH	jKS key store path. Mandatory if `Enable_SSL` is *true*.
Export KEY_PASSWORD	KS key store path. Mandatory if `Enable_SSL` is *true*.
Export KEY_ALIAS	JKS key store alias. Mandatory if `Enable_SSL` is *true*.
Export JMF_HOME	Refer the directory path created in Step 16. The Job Management Framework is responsible for executing loading jobs submitted via the ProSource Web UI.
Export CACHE_IN_MEMORY	Set to *true* to enable sever side memory loading of data source cache resulting better performance for pages.
Export GENERATE_ATTR_VALUE	If set to *true*, then for logs loading the dataset id the value is autogenerated.
Export DOCU- MENT_ENCODING	Set the encoding value based on your data, as required for document viewer and document scanner. For example, cp1251
Export JAVA_HOME	Path for java openjdk required in microservices. `<path to jdk-11>` Download `openjdk 11.`
Export TOPIC_SERVICE_SESSION_TIMEOUT	Time in seconds after which a topic service session expires releasing the unused license.
Export DROPDOWN_VALUES_ LIMIT	Data in the drop-down filter on the Collection's pages is limited by this value. Set this to zero to fetch all the data.
Export PROSOURCE_PYTHON_PATH	Python executable is installed as part of the ProSource installation. It is path to the Python executable within ProSource as mentioned in the comments section of the `sourceMS.config` file.
Export PROSOURCE_TOPIC_SERVICE_URL	As part of the URL, the hostname should always be the ProSource server hostname.

Action	Description
Export PYTHON_PATH	Python executable is installed as part of the ProSource installation. It is path to the Python executable within ProSource as mentioned in the comments section of the `sourceMS.config` file.
Export MICRO_SERVICE_PATH	This is micro services root folder from where all the micro-services are located.
Export MS_PROTOCOL	The field is either http or https depending ENABLE_SSL= *false* or *true* in PROXY SERVICE
Export SCANNER_FILE_PURGING_ TEMP_DIR	This property defines the directory path where the files deleted from ps web dropbox directories will be placed.
Export TESSDATA_PREFIX	This property defines the location of tessdata executables of Tesseract software deployed on user's environment. Export the path up to tessdata folder.
Export MAGICK_HOME	This property defines the location of the ImageMagick software installed on the user's environment.
Export LD_LIBRARY_PATH	This property exports the libraries of Image magick software. For example, export `LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$MAGICK_HOME/lib`
Export WAND_MAGICK_LIBRARY_SUFFIX	This property is a pre-requisite for image magick software. For example, export `WAND_MAGICK_LIBRARY_SUFFIX="-7.Q32;-7.Q32HDRI;.Q32HDRI;.Q32`
Export SEARCH_CONTENT_C RON_EXPRESSION	By default value of `SEARCH_CONTENT_CRON_EXPRESSION` will be - which indicates never run. If scheduling to be run, cron expression should be given as per the requirement.
Export EXT_CONTENT_SEARCH_DIRS	Search Content in view Data will search the contents from the directories mentioned at this path `EXT_CONTENT_SEARCH_DIRS` and `prosource external` directory
Export DASHBOARD_CACHE_ CRON_EXPRESSION	Cache for datasource can be built in the background with the help of cron expression. *cron* expression understands the expression like this *minute hour dayofmonth month dayofweek*.
Export INGESTION_REQUIRED	It should be set to *true* if *Export To Delfi* feature is required otherwise false.

Action	Description
Export DATA_DESCRIPTOR_CREATE_RECORD_END POINT	Please log on to the **DELFI** portal. Go to `API Catalog >> Storage Service >>` Create or update records. Use the given endpoint
Export INGESTION_SUBMIT_ ENDPOINT	Please contact DELFI administrator for Ingestion submit API endpoint to ingest on-prem data to cloud
Export UPDATE_SOURCE_ENTITY_ENDPOINT	Please log on to the DELFI portal. Go to `API Catalog >> Data Catalog Service >>` Update Source-Entity view. Use the given endpoint.
Export CREATE_RAW_ENTITY_VIEW_ENDPOINT	Please log on to the DELFI portal. Go to *API Catalog >> Data Catalog Service >>* Create `Raw/WKE Entity View`. Use the given endpoint
Export GENERATE_TOKEN_EN DPOINT	Request the DELFI administrator to supply the Endpoint to get the sauth token API. For example, `https://abscde.com/v1/svctk?key=`.
Export CRSPERSISTABLE_REFERENCE_NONWGS84_ ENDPOINT	Request the DELFI administrator for the API Endpoint to get the persistable CRS for ProSource projects other than WGS84.
Export UNITPERSISTABLE_REFERENCE_ENDPOINT	Request the DELFI administrator for the API endpoint to get the UnitPersistable Reference. The example below shows the endpoint for a Data Ecosystem P4D environment.
Export DELFI_AUTHORIZATION	User auth token used for ingestion. Log on to the DELFI portal, and go to the API Catalog section. Open any service and get the user token. Example: DELFI_AUTHORIZATION=Bearer *eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Ik1UVTVNVEUxT1Rnek1nPT0ifQ.eyJzdWIiOiJydmFsb3*
Export APPKEY	Used to access the authorized applications. Log on to the **DELFI** portal and go to the API Catalog section. Open any service and get the app key.
Export STATUSENDPOINT	Log on to the DELFI portal. Go to API Catalog > Ingestion Service > Status. Use the given endpoint. For example, `https://abcdexyz.com/status?jobId=`
Export ACCOUNT_ID	Use the tenant for which user has access to ingest data.
export KIND_VERSION	Used to define the version of the data ingested.

Action	Description
Export DELFI_SOURCE	Source of the data.
Export ACL	Data ecosystem DELFI administrator. For example, `{\"acl\": {\"viewers\": [ \"viewers@abcde.com\"], \"owners\": [ \"owner@abcde.com\"]}}`
Export LEGALTAGS	For example, `{\"legal\": {\"legaltags\": [ \"abc\"], \"otherRelevantDataCountries\": [ \"def\" ] }}`
Export DASHBOARD_READ_TIMEOUT	This value is Dashboard service to graph controller service timeout in milliseconds. The user can increase this Timeout value if the user observes any timeout issue while fetching the data for Smart Cache (Beta).
Export SLB_DATA_PARTITION_ID	It is a mandatory field. The ID of the partition you have assigned need to get from the DELFI administrator. Mostly, it is the same value as ACCOUNT_ID.
Export INGESTOR_SERVER_PORT	It is the port where the Ingestor Service will run.
Export SDUTIL_PATH	This is full file path to the sdutil command binary. The sdutil is installed by default when the ProSource installer is executed (in either fresh install or upgrade mode).
Export SDUTIL_PYTHON_PATH	This is the full path to the dedicated Python binary virtual environment used by sdutil.
Export GOOGLE_BUCKET	Provide the GCP path where the data file will be copied.
Export DATA_FILE_FULL_PATH	Comma separated list of Seabed database column names having any data file path. From those paths, the ingested files are stored into GCP or datastore before metadata ingestion. For example, *PATH, PATH1, PATH2*
Export SERVICE_ACCOUNT_A UTH_REQUIRED	Set to *true* or *false*. The user needs to contact the DELFI administrator to create service account and provide the required details.
Export SERVICE_ACCOUNT_I D	Contact your DELFI administrator for the value.

Action	Description
Export SERVICE_ACCOUNT_SECRET	Get this secret from DELFI and encrypt this using the utility EncryptionUtility.sh available at location`$PS_HOME/conf/dropbox/EncryptioUtility`. **Note:** It is recommended to rotate/ change the account secret within 90 days.
Export REFRESH_TOKEN_PROJECT_ID	Contact your DELFI administrator for the value.
Export TARGET_PROJECT_ID	Contact your DELFI administrator for the value.
Export TARGET_SERVICE_ID	Contact your DELFI administrator for the value.
Export AUTH_TOKEN_KEY	Contact your DELFI administrator for the value.
Export SEISSTORE_TENANT	Please contact your DELFI administrator to get the Seismic Store tenant root folder.
Export SEISMIC_SUBPROJECT	Optional. This is a subdirectory created below the SEISSTORE_TENANT path where seismic files are uploaded through sdutil.
Export SEISTORE_FILE_TYPE	The extension of the file type for the files to be uploaded through sdutil. Ensure that you properly find all the different extensions of the files originally loaded into ProSource Seismic.
Export MAP_SERVER_PROTO	Internet protocol of the ArcGIS server used for Map View, could be either http or https.
Export MAP_SERVER_HOST	Server hostname of ArcGIS server. $MS_HOST if ArcGIS server is on same machine where microservices are deployed.
Export MAP_SERVER_PORT	Port used by the ArcGIS server. Generally, this is 6443 for the https protocol and 6080 for the http protocol.
Export SAUTH_SSO_ENABLED	This property should be set to "true" only if SSO needs to be enabled via DELFI ecosystem. **Note:** Either SAUTH SSO Enabled OR AZURE_SSO_ENABLED can be made true at a time.
Export AZURE_SSO_ENABLED	This property should be set to *true* only if SSO needs to be enabled via Azure Active directory

Action	Description
Export SAUTH_CLIENTID	This is the SAUTH client ID (can be obtained after registering application on SAUTH portal). **Note:** Get encrypted secret keys from this Utility at location `%PS_HOME%/conf/dropbox/EncryptioUtility`
Export SAUTH_CLIENT_SECRET	This is the SAUTH client Secret (can be obtained after registering application on SAUTH portal). **Note:** Get encrypted secret keys from this Utility at location `%PS_HOME%/conf/dropbox/EncryptioUtility`
Export SAUTH_AUTHENTICATE_URL	This property is for Delfi or Azure Authentical URL.
Export SAUTH_TOKEN_SERVICE_URL	This property is for Delfi or Azure Token Service URL.
SAUTH_AUTHENTICATION_SERVICE_ENDPOINT	This Property is required for Azure only. Uncomment and set this value as per your Azure App registration for Azure SSO login.
SAUTH_TOKEN_SERVICE_ENDPOINT	This Property is required for Azure only. Uncomment and set this value as per your Azure App registration for Azure SSO login.
DB_DATE_FORMAT	Represents date format of date data retrieved from DB.
DB_DATE_TIMEZONE	Represents application server timezone. The timezone name can be retrieved by running timedatectl on the server.
LOGS_SERVICE_READ_TIMEOUT	Ribbon timeout specific to logs-service in milliseconds. Increase this value if file download fails from ps web due to timeout.
COMPRESSION_REQUIRED	Set this to true to compress datasource cache build pages and collections
DASHBOARD_SCHEDULER_THREAD_COUNT	Increase this count to do more parallel processing of datasource cache.

Note: Actual path of files may vary from the examples in the commented section of the `sourceMS.config` file.

Save the `sourceMS.config` file.

11. Install Python virtual environments.

ProSource Web uses Python enabling users to use Python capabilities. Download 2023.1 python-modules.zip from the SLB SDC site, copy in `$MS_HOME` and extract it using the command unzip -o python_modules.zip. The following features of ProSource Web are based on Python:

- SEG-Y auto scanner
- SEG-Y full scanner
- Dashboards
- Documents scanner
- OSDU Nifi
- Sdutil

All the features are run on separate Python virtual environments, as each virtual environment has a different and conflicting set of Python packages for the respective utility. To use these features, the ProSource administrator must do a one-time setup of those virtual environments by executing *venv_installation_script.sh sourceMS.confing* in `$PS_HOME/microservices`. This process may take a few minutes to finish.

Go to `$PS_HOME/microservices/python-modules` to check the virtual environment created with a folder for each virtual environment.

12. To start microservices, execute the `./startMS.sh` command.
13. To check the status of all the microservices, execute the `./startMS.sh -s` command.

Note: Graph controller services are obsolete in this release and will be removed from the microservices list in the next major release so will remain in stop status only.

```
DISCOVERY                       : Running
JMF                             : Running
PROXY                           : Running
AUTHENTICATION                  : Running
CONTROLLER                      : Running
CONTENTSEARCH                   : Running (CONTENT_SEARCH_REQUIRED=true)
SCANNER                         : Running
SEISMIC SERVICE                 : Running
LOGS SERVICE                    : Running
EXTRACTOR                       : Running
KAFKA                           : Running
ZOOKEPER                        : Running
DASHBOARD                       : Running
EXPORTER                        : Running
JOB MANAGER                     : Running
TRANSFER                        : Running (INGESTION_REQUIRED=true)
INGESTOR                        : Running (INGESTION_REQUIRED=true)
TOPIC                           : Running
Graph controller                : Stopped
Email service                   : Running
Report service                  : Running
OSDU INGESTOR                   : Running (OSDU_INGESTION_REQUIRED=true)
OSDU AUTH TOKEN                 : Running (OSDU_INGESTION_REQUIRED=true)
Plugin Service                  : Running
RATING SERVICE                  : Running
```

14. To stop all or any desired service then run the `./stopMS.sh` command and provide the respective service option to stop from the usage list.
15. Launch the ProSource web using the URL in the *http(s):///<MS_HOST>:< PROXY_PORT>* format.

Note: To get the details of an individual service, check the microservice log file created at the location `$PS_HOME/microservice/logs`.

Data Ingestion to OSDU

The ProSource to the OSDU Data Platform is a set of ProSource microservices which can be used to ingest the data loaded in the ProSource database to OSDU. We can ingest Seismic (SEGY and ZGY), Logs (LAS and DLIS), and CSV data from various topics into OSDU.

Note: The OSDU configuration and Nifi deployment are required for data ingestion from ProSource to OSDU. For more details, please refer to the ProSource to OSDU data ingestion section of the Online Help in the ProSource Learning Center

DataSource Caching for Faster Performance in Pages and Collection

To fetch the data faster in pages and collection the caching mechanism is provided.

Cache for datasource can be built in the background with the help of cron expression.
To Configure Cron job for datasource cache make following settings:

1. In sourceMS.config file define value for property DASHBOARD_CACHE_CRON_EXPRESSION like
   `export DASHBOARD_CACHE_CRON_EXPRESSION="* 0/10 * * *"`
   Cron expression understands the above expression like this *minute hour dayofmonth month dayofweek*.
2. Configure project, interfaces, datasources, and users for which you want the cache to be created at every scheduled interval in the following configuration file.
   `$PS_HOME/microservices/prosource-dashboard/dash_directory/CacheConfig/config.json`
   For non-entitled interface and projects set *users* value as `ALL_USERS` and `isEntitled` values as false.
   Based on the configuration for any individual project or interface, cache should be created.
   a) If isEntitled the property is set to true, and groups are defined then the cache should be created per group per datasource per project or interface and only a list of groups configured for that Project or interface would be eligible for automatic cache creation on the Scheduled time. If any user is present in multiple groups then user level cache will be generated for that user (no group level cache for that user). If there are no valid users in the group then group level cache will not be generated.
   b) If `isEntitled` property is set to false, then cache should be created per datasource per project or interface and shared among all users
   c) If `isEntitled` property is set to true, so If any new user comes or any new datasource is added in ProSource which is not configured as a part of cache config, the cache should be created on the fly when first requested
3. Parallel processing for the cache is supported and the number of thread counts for processing can be controlled by `DASHBOARD_SCHEDULER_THREAD_COUNT` present in sourceMS.config under microservices. This can be increased depending on the number of cores present on the server on which ProSource is hosted. Restart the dashboard service.

Configuration for delta updates in pages

This feature improves the performance of loading pages and collection by enabling delta updates or refresh of cache when any new data is loaded. To enable this feature, the prerequisites and configuration below are required in PS Web.

Mandatory columns:

Data sources must have Insert Date and Update Date as a display name. Only the pages created on top of data sources that fulfill these criteria are eligible for incremental delta updates otherwise a full refresh of the cache is carried out.

Note: Whenever any data source definition is modified, kindly ask the ask users to run db sync on the existing pages based on that data source. If you are creating new pages, then perform a sync them before adding any widgets. If the attribute display name is modified in the data source definition, then for the existing pages based on that data source the widgets may not work if they are using modified attribute.

Set Unique Key:

Admins can select one or multiple columns as a unique key in a data source definition on the **Manage Data sources** tab. If no unique key is set for a data source definition, then the pages created on top of that data source are not eligible for incremental delta updates, instead a full refresh is done.

If an Admin selects single or multiple columns as a Unique key in a data source definition, and if the data based on the selected unique keys does not provide unique data, then the pages created on top of that data source are also not eligible for incremental delta updates, instead a full refresh occurs.

Set ID attributes: There should be one ID attribute set in the data source definition.

Uploading Large Files

The default file upload limit in ProSource Web is 3 GB.

1. To configure large file upload, modify value for below variables in sourceMS.config as required.

   export `UPLOAD_FILE_REQUEST_LIMIT_GB=11`

   export `RIBBON_READ_TIMEOUT=1200000`

   To upload a 10 GB file, configure it to 11 GB, 1 GB more than the actual file size, and upload timeout as 1200000 ms (milliseconds) or as per your upload size limit.
2. Restart Proxy and Scanner Services to take effect.

Note: `UPLOAD_FILE_REQUEST_LIMIT_GB` variable is for individual file size and not for overall files upload size

Mandatory attributes for logs loading

This is an optional configuration. Admins can configure any desired attributes as mandatory and change the attribute order for the Logs Loader queue. This configuration can be done using the file `logsloading.configuration.json` available at `$PS_HOME/microservices/`

`prosource-configuration-services/config/`. For more information, refer to the Learning Center.

Document Scanner Configuration

Configure the document scanners in the dropbox.config file to scan and extract metadata from Documents such as *PDF/Images/TIFF/Office* format files.

You can add the type of extensions you would want to scan.

For extensions marked *scannable=true*, is picked up by the document scanner for metadata extraction.

```
<scanner class="com.slb.sis.seisstream.job.dropbox.pollers.datafile.DocumentScanner" name="DOCUMENT-JSON">
    <extensions>
        <extension scannable="true" value="png"/>
        <extension scannable="true" value="jpeg"/>
        <extension scannable="true" value="jpg"/>
        <extension scannable="true" value="tif"/>
        <extension scannable="true" value="tiff"/>
        <extension scannable="true" value="doc"/>
        <extension scannable="true" value="docx"/>
        <extension scannable="true" value="ppt"/>
        <extension scannable="true" value="pptx"/>
        <extension scannable="true" value="xls"/>
        <extension scannable="true" value="xlsx"/>
        <extension scannable="true" value="pdf"/>
    </extensions>
```

After adding all the required extensions, make the below respective configurations to the document scanner properties. All the paths must be the absolute path (replace $PS_HOME with the absolute path in the dropbox.config file).

```
<properties>
<entry>
<key>PYTHON</key>
<value>$PS_HOME/microservices/python-modules/doc_scanner_env/ bin/
python3.9</value>
</entry>
<entry>
<key>PYSCRIPT</key>
<value>$PS_HOME/microservices/scanner_standalone/ document_scanner/
DocumentsScannerPackage/ run_documents_scanner.py</value>
</entry>
</entry>
<key>TIKA_CLIENT_PATH</key>
<value>$PS_HOME/microservices/scanner_standalone/ document_scanner/
3rdpartyjar/tika-app-2.3.0.jar</value>
</entry>
</entry>
<key>ENABLE_CONTENT_EXTRACTION</key>
<value>TRUE</value>
</entry>
</entry>
<key>SPACY_MODEL_PATH</key>
<value>$PS_HOME/microservices/python-modules/doc_scanner_env/ lib/
python3.9/site-packages/en_core_web_lg/en_core_web_lg- 3.2.0</value>
</entry>
<entry>
<key>MODEL</key>
<value>$PS_HOME/microservices/scanner_standalone/ document_scanner/
```

```
DocumentsScannerPackage/models/ documents_scanner_default_model</value>
</entry>
<entry>
<key>ENABLE_DOCUMENT_CLASSIFICATION</key>
<value>TRUE</value>
</entry>
<entry>
<key>DOCUMENT_CLASSIFICATION_MODEL</key>
<value>$PS_HOME/microservices/scanner_standalone/ document_scanner/
DocumentsScannerPackage/models/ documents_classification_default_model</
value>
</entry>
<entry>
<key>TESSERACT_PATH</key>
<value>/home/install/Downloads/tesseract/bin/tesseract</ value>
</entry>
<entry>
<key>SIMILARITY_BUFFER</key>
<value>0.1</value>
</entry>
<entry>
<key>PAGE_SEGMENTATION_MODE</key>
<value>4</value>
</entry>
<entry>
<key>BEST_MATCHES</key>
<value>4</value>
</entry>
<entry>
<key>ENABLE_TABLE_EXTRACTION</key>
<value>TRUE</value>
</entry>
<entry>
<key>CONTENT_PAGES</key>
<value>1-10</value>
</entry>
<entry>
<key>IMAGE_RESOLUTION</key>
<value>200</value>
</entry>
<entry>
<key>NUMBER_OF_THREADS</key>
<value>10</value>
</entry>
<entry>
<key>SECTION_SIZE</key>
<value>50</value>
</entry>
<entry>
<key>DATA_EXTRACTION_CONTROLFILE_PATH</key>
<value>$PS_HOME/microservices/scanner_standalone/ document_scanner/
DocumentsScannerPackage/models/ data_extraction_default_controlfile</
value>
</entry>
<entry>
<key>PARALLEL_EXECUTION_COUNT</key>
<value>4</value>
</entry>
<entry>
<key>MAX_TIMEOUT_MINUTE</key>
```

```
<value>1</value>
</entry>
</properties>
```

Document Viewers

There are some configurations required for documents of various extensions to work. All the standard document formats and their supported viewers have been listed in the configuration example below:

```
<viewer>
    <extensions>
        <extension type="txt_array" value="dat"/>
        <extension type="txt_array" value="txt"/>
        <extension type="txt_array" value="csv"/>
        <extension type="txt_array" value="las"/>
        <extension type="txt_array" value="json"/>
        <extension type="txt_array" value="p190"/>
        <extension type="txt_array" value="sps"/>
        <extension type="txt_array" value="p90"/>
        <extension type="txt_array" value="uko"/>
        <extension type="txt_array" value="ukooa"/>
        <extension type="txt_array" value="p184"/>
        <extension type="txt_array" value="p111"/>
        <extension type="txt_array" value="xps"/>
        <extension type="txt_array" value="rps"/>
        <extension type="txt_array" value="nav"/>
        <extension type="txt_array" value="p176"/>
        <extension type="img_array" value="png"/>
        <extension type="img_array" value="jpeg"/>
        <extension type="img_array" value="jpg"/>
        <extension type="img_array" value="gif"/>
        <extension type="img_array" value="bmp"/>
        <extension type="img_array" value="svg"/>
        <extension type="img_array" value="webp"/>
        <extension type="img_array" value="tif"/>
        <extension type="img_array" value="tiff"/>
        <extension type="office_array" value="doc"/>
        <extension type="office_array" value="docx"/>
        <extension type="office_array" value="ppt"/>
        <extension type="office_array" value="pptx"/>
        <extension type="xls_array" value="xls"/>
        <extension type="xls_array" value="xlsx"/>
        <extension type="pdf_array" value="pdf"/>
    </extensions>
</viewer>
```

File Upload

Only the predefined type of file formats is applicable to be uploaded into the server root directory folders via PS Web. This ensures a more secured way of dropping files into the server via the application. Make sure that all the file format extensions declared under the individual scanners are automatically recognized and only specific file formats that are uploaded upon the administrator's approval.

Make sure that all data files are scanned with antivirus software to negate virus and malware threats.

```
<upload>
    <extensions>
        <extension value="xml"/>
        <extension value="json"/>
    <extension value="pds"/>
    </extensions>
</upload>
```

Search within the data file configuration

Warning: This feature is in beta state and currently doesn't honor user entitlement while returning results. Their feature requires high RAM usage(more than 128 GB) and if system RAM is less then sometimes systems run out of memory and cause other services to stop.

This search functionality enables users to search the files that contain the text being searched. Files added in the configured directories on Files and Folders of Scan and Load features. Users can also configure external directories from where the files can be searched using the content text in the file.

1. From SDC download ProSource 2023.1 - ElasticSearch for File Content Search feature and extract the elastic search installer elasticsearch- 6.8.14.tar.gz at any suitable location. The command:

   `tar -xvzf elasticsearch-6.8.14.tar.gz`
2. Configure the following properties in `sourceMS.config`.
   a) List of folders for the variable on which the search within the data file fetches the files based on search text. A single folder path is allowed. Multiple paths can be provided with comma (,). For example, export EXT_CONTENT_SEARCH_DIRS=/home/install/Music,/apps/ data_files

   Note: There is no need to add PS_EXT or any drop-box directories as these are indexed by default. (if one is added then this may lead to incorrect search results).

   b) To run the files indexing after regular interval (for example, every 3 mins in below settings), export SEARCH_CONTENT_CRON_EXPRESSION="0 0/3 * * * *"
   c) Export `JAVA_HOME=<path to jdk-11>` and download openjdk 11 from `https://download.java.net/java/ga/jdk11/openjdk-11_linux-x64_bin.tar.gz`.
   d) export `ELASTIC_SEARCH_PATH=<path to folder where elastic search is extracted>` exclude folder name elasticsearch-6.8.14 from path.
3. In the files startMS.sh, stopMS.sh, and venv_installation_script.sh, search and set the below flag as true:

   `ELASTIC_SEARCH_SERVICE_FLAG=true`

   and rebuild python environment.
4. Restart the Scanner, Controller, and Content Search service to reflect the changes.

Note: If the newly copied files in configured directories are not displayed in the search result, the files may have an old timestamp than the last indexing run timestamp. In such cases, you might have to touch these files to modify the timestamp and get

it indexed. To modify the timestamp, use the linux command: touch `/apps/data/files/LogsData.pdf`.

Set Templates Priority for the Seismic Auto- Scanner

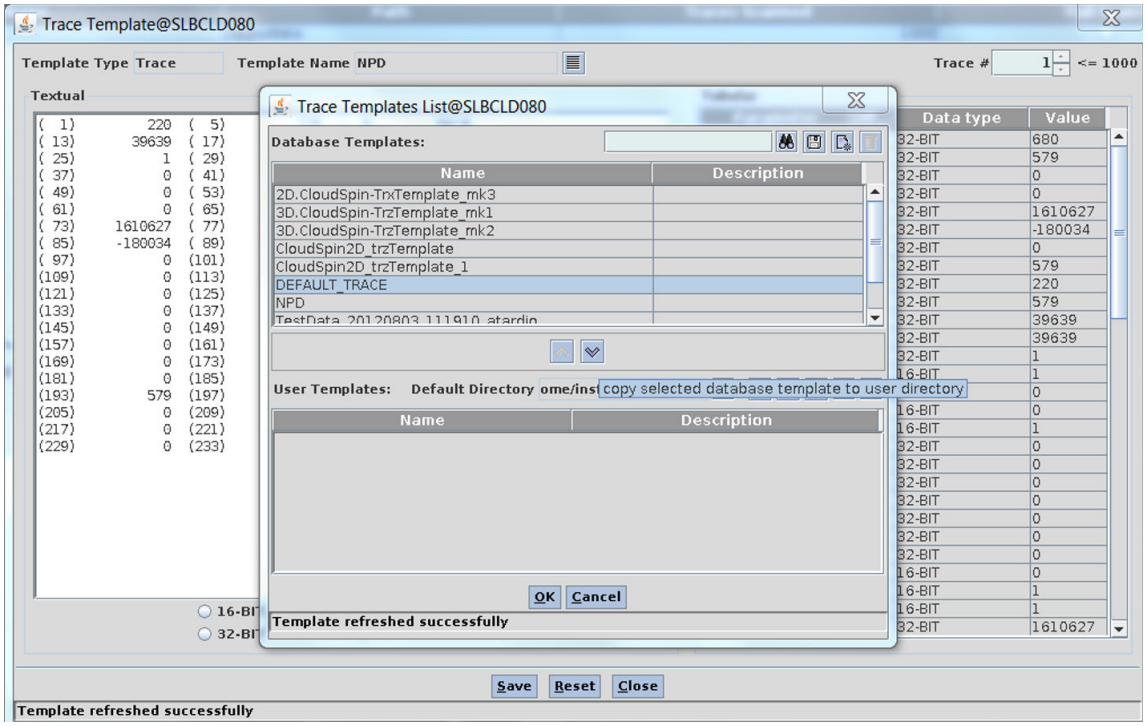
The Auto-scanner program runs as part of the crawling process and predicts the following properties for a SEG-Y file:

- Primary Sort Key
- Stack Type
- Survey Type
- Best suited trace header template

The auto-scanner program iterates through a list of trace header templates and determines the most suitable one for the SEG-Y file being crawled.

In order to create this library of template, perform the following steps:

1. Open Scan/Edit/Load and select data format as SEG-Y, select any output directory and then select **Next**.
2. Add any SEG-Y file.
3. Select **Trace Header Template** and open the **Create/View** template window.
4. Copy the templates to the desired directory.



5. In the `dropbox.config` file, fill the value of Template Archive as the directory where templates were copied in the step above.
6. Set the priority of templates. The auto-scanner iterates and predicts the parameters by first picking templates from this list.

Note: This is a mandatory step.

```
            </value>
        </entry>
        <entry>
            <key>TEMPLATE_ARCHIVE</key>
            <value>/home/install/Templates</value>
        </entry>
        <entry>
            <key>TEMPLATE_PRIORITY</key>
            <value>SEGY_REV_1_SP197_SPSCALAR201.sgyfmt,Big_File_2D.sgyfmt,Big_File_3D.sgyfmt,SEGY_REV_0.sgyfmt,SEGY_REV_1.sgyfmt</value>
        </entry>
    </properties>
</scanner>
```

Deploy Microservices on a Remote Server

ProSource and Tomcat are setup on one server and only microservices are deployed on the other server. Microservices uses the database connected to ProSource.

Important: The following step is to be followed on the ProSource server.

1. Go to `$PS_HOME/microservices/prosource-jmf` on the ProSource server and do the following:
   a) Run the `./jmf_make.sh` command.
   b) Select option 2, 3 or 4, to prepare JMF zip for the respective loading services as shown in the image below. The new ZIP file is created in the same location where the jmf_make.sh script is present.

```
JMF Install Script
Enter 1  Install JMF Locally
Enter 2  Prepare JMF zip for Seismic and Well data Loadi
Enter 3  Prepare JMF zip for Well data Loading
Enter 4  Prepare JMF zip for Seismic data Loading
Enter 5  Install JMF
Any other option to Exit
```

Important: The following steps are to be followed on the Remote server.

Important: Stop all microservices if already deployed from the correct location.

2. Install Oracle Client 19c (runtime version).
3. Configure `tnsnames.ora` files under `<ORACLE_HOME>/network/admin` to point to the ProSource database.
4. Create a new directory on the remote server to install microservices. For example, MS_REMOTE.

Note: For reference purposes, consider this directory path as MS_REMOTE throughout the documentation below.

5. Copy the microservices folder from `<PS_HOME>` in the ProSource server to the `MS_REMOTE` directory.

   With this zip file created in Step 1 also gets copied in the respective folder.
6. Create a dummy folder path `/ext_app` environment on your **MS_REMOTE** machine which resembles to the original `<PS_HOME>/ext_app` path.

7. Copy the activemq folder from `<PS_HOME>/ext_app` of the ProSource server to the `MS_REMOTE` directory.
8. Go to `$PS_HOME` on original server and copy below files to the dummy `$PS_HOME` folder on the remote server.

   keyPass.dat

   keyPassCpp.dat

   encryptKeyDbCpp.jceks

   encryptKeyDb.jks
9. Install ProSource and Oracle dependencies.

   Execute from the root user "yum install compat-db47 compat- libf2c-34 compat-libstdc++-33 compat-openldap cyrus-sasl- lib elfutils-libelf expat fontconfig freetype glibc keyutils-libs krb5-libs libICE libSM libX11 libXau libXaw libXdamage libXext libXfixes libXft libXi libXmu libXp libXrender libXt libXtst libXxf86vm libaio libcom_err libdrm libgcc libjpeg-turbo libpciaccess libpng libsasl2 libselinux libstdc++ libstdc++-devel libuuid libxcb mesa- dri-drivers mesa-filesystem mesa-libGL mesa-libGLU mesa- private-llvm motif ncurses-libs nspr nss nss-softokn nss- softokn-freebl nss-util openldap openssl readline sqlite tcsh xorg-x11-xbitmaps xterm zlib binutils compat-libcap1 compat-db compat-libstdc++-33 control-center gcc gcc-c++ glibc glibc-common glibc- devel glibc-headers ksh libaio libaio-devel libgcc libgnome libgnomeui libgomp libstdc++ libstdc++-devel libXp libXtst"

   Note: Install RPM 'dos2unix' which is not part of the standard Redhat packages.

   Note: Ensure the hostname of the ProSource server and ProSource license server are resolvable. For example, "ping <ProSource hostname> on Linux terminal".
10. Mount the ProSource external directory. For example, `/apps/sis/PS_EXT` on to the remote server where microservices are deployed.
11. To install JMF on remote setup, unzip the zip file created in Step 1 or copied in Step 5 using the unzip `<PS_JMF_Folder.zip>` command from location `<MS_REMOTE>/microservices/prosource-jmf`

    Export `ORACLE_HOME` on the remote server. For example,

    Export `ORACLE_HOME=/apps/oracle/19c/product/19.3.0/client_1`
12. Go to the PS JMF folder and make scripts executable using the chmod a+x*.sh command.
13. Run the `./jmf_make.sh` command. Run dos2unix jmf_make.sh in case of any issues.
14. Select **Option 5** as shown in the image below.

    ```
    JMF Install Script
    Enter 1  Install JMF Locally
    Enter 2  Prepare JMF zip for Seismic and Well data Loading
    Enter 3  Prepare JMF zip for Well data Loading
    Enter 4  Prepare JMF zip for Seismic data Loading
    Enter 5  Install JMF
    Any other option to Exit
    ```
15. Enter the complete path in the `MS_REMOTE/<JMF_Directory name>` format. A directory is created by script automatically if it is not already present.

    Note:

For reference purposes, consider MS_REMOTE/jmf_remote as <JMF_HOME> throughout the documentation below.

16. Now go to the `/apps/MS_REMOTE/PS_EXT/tomcat/dtmgui/work/control/ templates` location on your **MS_REMOTE** environment and then update the respective loader and exporter template files by replacing the `<PS_HOME>` path with `<JMF_HOME>/`resources path.
    MapFile and CoralRootPaths for most template files
    Before: `CoralRootPath= /apps/sis/prosource/PS2021/ext_app/psx/Seabed_Utilities/config`

    After: `CoralRootPath=/apps/MS_REMOTE/jmf_remote/resources/conf/tahiti/coral`

    Note: Do not mount the ProSource home directory from the ProSource server on to the remote server where microservices are deployed.

17. Refer to step 5 onwards from the section *Deploy Microservices on the ProSource Server (Locally)* for the remaining configuration and do changes in corresponding files on the remote server.

ProSource additional configuration and procedures

ProSource installation involves some additional configuration steps which are outlined in this section.

Configuration of SSL LDAP for ProSource

Follow the steps below to configure SSL LDAP for ProSource

1. Request an SSL certificate from your organization.
   For example, for SLB you can request a certificate and download the SLB Certificate keychain from below URL.
   https://certificate.slb.com/certsrv/
2. Include above requested certificate in cacerts keystore file present at:
   `PS_HOME/jre64/lib/security` (a default password is changeit).
   You can use windows GUI tool such as Keystore Explorer for working with keystores.
3. Go to `$PS_HOME/TomcatHome/webapps/imadmin` and open **IMADMIN** console by using `./safe-start-imadmin.sh`
4. Edit **UserMangementAD** and set below details:
   LDAP Provider SSL URL: ldaps://dir.slb.com:636
   LDAP Security protocol: ssl
   Trusted Keystore location: `$PS_HOME/jre64/lib/security/cacerts`
5. Restart Tomcat, ProSource server and microservices.

Change SIS_ADMIN Password

1. Stop all microservices, ProSource server and Tomcat server.
2. Change **sis_admin** password in java.
   ```
   cd $PSHOME/conf/dropbox/EncryptioUtility
   java -jar EncryptionUtilityJar.jar -DstoreAdminPassword="<new_password>"
   ```
3. Change **sis_admin** password in cpp
   ```
   cd $PSHOME/ext_app/pss/scripts
   ./encryption_utility.csh "<new_password>"
   ```
4. Change password in Oracle and run procedure to change in legacy
   ```
   sqlplus / as sysdba
   exec sds_sys.sds_admin.sisadmin_password('<new_password>');
   alter user sis_admin identified by <new_password>;
   ```
5. Start Tomcat, ProSource and microservices.

Log in to Oracle account using DB token

Using the dropboxUtility.sh, you can get a password which is used to connect to DB via SQL*Plus or SQL Developer. To login to oracle account, proceed as follows:

1. Go to `$MS_HOME` and open `sourceMS.config`.
2. Under **Drop box and DB token** utility, search for `AUTHORIZATION_URL`.

   By default you should have AUTHORIZATION_URL= http:

   `$MS_HOST:$AUTHORIZATION_PORT` located under **Drop box and DB token** utility. Change this to localhost from $MS_HOST and save the file.
3. Go to `$PS_HOME/util/dropbox and execute ./dropboxUtility.sh`.
4. Select DB Token utility.
5. Enter the username and password.

   It will generate password that is used to connect to oracle via SQL*Plus or SQL Developer.

Gunicorn configuration to improve performance on dashboard pages

The ProSource Dashboard Service uses Gunicorn for the serving http request.

There are certain settings for Gunicorn which can be defined in file start_dashboard.sh, present at `$PSHOME/microservices/prosource-dashboard`

- **NUMBER_OF_WORKERS**

  The number of worker threads for handling requests. Generally below formula is used to identify maximum number of workers that could be set:

  Number of workers = 2 * (No of cores) + 1
- **--worker-connections**

  The maximum number of simultaneous clients for a worker. By default, in ProSource it is configured to 2000.
- **--timeout**

  Workers silent for more than this many seconds are killed and restarted. This will help to release volatile memory and make it available for other workers and services.

  Value is a positive number or 0. Setting it to 0 has the effect of infinite timeouts by disabling timeouts for all workers entirely.
- **--max_requests**

  The maximum number of requests a worker will process before restarting.

  Any value greater than zero will limit the number of requests a worker will process before automatically restarting. This is a simple method to help limit the damage of memory leaks.

  Refer to Gunicorn online documentation present at *https://docs.gunicorn.org/en/stable/settings.html*

Example: If there are 10 users simultaneously loading a view data page which sends out 5 http concurrent requests to each user, then in total 50 concurrent client requests are to be served by Gunicorn.

So ideally to avoid any sequential delay in serving request, the factor number of workers should be as much close to concurrent http requests made.

However, in practical it is not possible to a have that number of cores so with lower configuration the parameter --worker-connections comes into play to handle requests.

For example: If each request takes 1 millisecond to handle, then a single worker will serve 1000 RPS. If each request takes 10 milliseconds, a single worker will serve 100 RPS.

If a request takes 10 milliseconds, others take, say, up to 5 seconds, then you'll need more concurrent worker, so the requests that takes 5 seconds can go on other worker and does not delay all other request.

Warning: Warning: If in file sourceMS.config the CACHE_IN_MEMORY is set as true then increasing the number of workers will ultimately require that many times more volatile memory for example, if earlier 1 worker required 2 GB RAM to process request, then with 5 workers it will be 2*5= 10GB to server concurrent request. In this case you can define --timeout and --max_requests parameters with lower values to minimize memory issues.

Configuration for Plugin transfer

The Transfer of Deviation Survey data will be in a disabled state by default and will not be transferred even though it is selected in the Techlog Transfer window.

1. To enable Deviation survey transfer, go to folder **ProSource for Techlog** For example, `C:\Users\current_user\AppData\Roaming\Schlumberger\Techlog\Extensions\Schlumberger\ProSource for Techlog\2020.1\2021.2.0.35\bin`and open the file `PSWeb.Infrastructure.Plugin.dll.config` and change the value to "true" for key="TransferDeviationSurveyToTechlog".
2. Save the file and restart **Plugin service** to see effect.

Note: Currently maximum transfer of 5000 entities (including child items) in one go is supported. If a user submits the transfer where sum up count of entities (including all parent and child items) are more than 5000 then those jobs may fail after partial transfer.

Troubleshooting

This section provides solutions to various common issues related to the ProSource server.

ProSource Installer reports zero space available for disks with 1+ TB free space

The ProSource installer will report zero space available for very large disks with free space greater than 1 TB and refuse to continue with the installation. The ProSource installer is implemented using a 3rd-party component named InstallAnywhere. A problem in this application causes it to report zero (0) disk space on large disks when the available free space is greater than 1 TB. This will also prevent the installer from continuing.

Solution

Set the environment variable `CHECK_DISK_SPACE` to the value `OFF` for example:%>setenv CHECK_DISK_SPACE OFF prior to running the ProSource installer.

Note: This should be used with caution and only if necessary as it will disable disk space checks and the installation will proceed in all cases even if there is insufficient space.

ProSource upgrade installer fails giving OpenLDAP error

If ProSource was configured with multiple LDAP URLs, the ProSource upgrade fails reporting errors about OpenLDAP not running.

Solution

In order to bypass this error, launch IM Administration Console in safe-start modeand edit the User Management component under the **Enterprise** tab. Remove the second URL from the LDAP provider URL parameter and save it. Now continue with the ProSource upgrade installation and just before the final install step, undo the changes done above, in IM Administration Console safe-start mode.

Recover from a failed/cancelled/aborted ProSource installation

If your ProSource installation fails, the following conditions may apply:

- OpenLDAP services fail to initialize.
- Oracle listener disconnected.
- There is insufficient swap space.

Uninstallation of ProSource version is not supported. Restore to the previous installation system backup or snapshot on both Oracle and ProSource server or if cold backup is taken then follow the steps below:

1. Delete the $PS_Home and PS_EXT directories tree from the ProSource server and restore its backup folders.
2. Revert to the old Seabed backup.
   a. Launch the ProSource installer.

b. Upgrade ProSource.

For more information refer to *Backing up Data* under the section *Seabed Upgrade Process Detail* in the *Seabed Project Upgrade Guide*.

Folders/dropbox directories do not display scanner/loader file explorer due to fewer threads

Example: If the Server has 16 Cores and configured directories are 12, then PS Web may not show files in File Explorer, and the scanner service may display an error in the service start log.

If this happens, update the configuration bekiw in `$MS_HOME/scanner_standalone/config/application.properties`

```
pss.crawler.watchService.numberOfThreads=2 <new value - 12>
```

numberOfThreads under `scanner config/application.properties` should be approximately 80% of the total CPU cores on the server if directories configured in `scanner config/dropbox.config` are more than the predefined numberOfThreads.

Microservices Related Common Issues

1. If any service is not running, make sure that the script has the proper execution rights. Since folders are being copied from the ProSource server to another server, there are chances that execution rights have been tampered.
2. If any script is being executed, then try running dos2unix on that script. For example, dos2unix jmf_make.sh.
3. While deploying microservices on the remote machine in the `sourceMS.config file`, export PROSOURCE_TOPIC_SERVICE_URL= must be set to the ProSource server URL since this accesses the topics from the ProSource server.
4. Open the proxy port to access the ProSource web from outside.

Job Definitions Restoring

To update the job_definitions file, follow these steps:

1. Go to `$PS_HOME/conf/tahiti`.
2. Open `job_definitions_backup.xml` and copy the custom settings to `job_definitions.xml`.

Please review the steps described in the *Post-Installation Tasks* section of the *ProSource Seismic Administration Guide* to make sure that the system is properly configured.

Seabed Extension export failure

Failure caused by: `java.sql.SQLException: ORA-29540: class com/slb/sis/sds/extend/ util/ExportDefinition` does not exist. If the Seabed extension fails with a similar error, then proceed as follows:

1. Go to Seabed home:`cd $SDS_HOME/install`.

2. Run the following Linux command:`loadjava -user sis_catalog/<sis catalog password>@<instance name> -thin -resolve ExportDefinition.class.`

   The above command needs `SIS_CATALOG` password and instance name to connect to the database.
3. Restart **Tomcat** and the **ProSource** server and retry extension export.

Seabed utility package CALC_WELL_POINT is not working

If the Seabed utility package CALC_WELL_POINT is not working, proceed as follows:

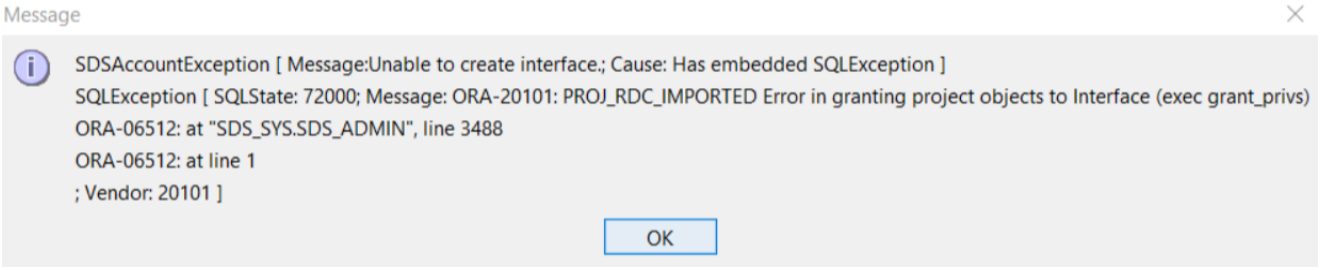
1. Go to directory `$SDS_HOME/lib`.
2. Run the following Linux commands for all the DD's:

```
 loadjava -resolve -user <DD account>/<DD account password>@<instance name> -grant PUBLIC SDSBulk.jar.
```

```
loadjava -resolve -user <DD account>/<DD account password>@<instance name> -grant PUBLIC SDSPSC.jar.
```

Interface creation on new project

The Interface creation on new project fails with the below error:



1. Navigate to `$PS_HOME/install/sp/postinstall` and execute below commands:
   - `dos2unix *.csh`
   - `chmod 755 *.csh`
   - `./Project_layer_update.sh`
2. Select option 1 and provide project name for which interface failed and its password and recreate new interface.