



Spectrum Spatial for Big Data

Version 5.0

Release Notes

Contents:

What's New?.....	2
Known Issues.....	3
System Requirements and Dependencies.....	5



What's New?

The Spectrum Spatial for Big Data version 5.0 suite of products contains the following new and updated features. For more information on all these updates, see the Spectrum Spatial for Big Data user guides on the [Spectrum Spatial for Big Data](#) documentation landing page.

Spectrum Geocoding for Big Data

New Operational Addressing SDK

This release includes a new SDK that provides APIs for Spark integration for the following operations:

- **Verify:** Accepts an address as input and returns the standardized US or international address and additional attribution for the address.
- **Geocode:** Accepts an address as input and returns the standardized US or international address and additional attribution, including the location coordinates, for the address.

For more information, see the *Spectrum Geocoding for Big Data User Guide* on the [Spectrum Spatial for Big Data](#) documentation landing page, as well as the Spark Addressing API Scaladocs available in your distribution:

```
/pb/addressing/software/spark2/sdk/scaladocs/bigdata
```

Databricks Geocoding Sample

A new sample for Databricks demonstrates how to install, configure, and execute a geocoding process in Databricks. The Geocoding Installation notebook files and instructions are provided on GitHub:

<https://github.com/PreciselyData/big-data>.

Hadoop Distribution and Framework Support

New

- Added support for Cloudera 5.16, 6.3 and 7.1

Library Upgrades:

- Spectrum Global Geocoding SDK (GGS) 4.0.2
- Global Routing SDK (GRA) 2.0.12
- Location Intelligence SDK (LI SDK) 1.14.1

Known Issues

- **Geocoding SDK:** When using the Geocoding CLI tool to manually configure your reference data, the bash command now requires an additional flag: `--t json`. This command generates the `JsonDataConfig.json` file, which contains configuration information for your geocoding datasets and needs to be regenerated for every software and data release.

```
cli configure --t json --s /precisely/geo_addr/data --d /precisely/geocoding/software/resources/config
```

It is strongly recommended to use automatic data configuration, by providing the location of the data, to configure the geocoding engine. For more information, see the *Spectrum Geocoding for Big Data User Guide* on the [Spectrum Spatial for Big Data](#) documentation landing page.

- (HAD-6867) **Addressing SDK:** Some fields on the result object may be inaccessible using an output field expression. If you see an error that starts with 'Error accessing field: Unable to invoke method', you may need to use a custom executor to access that method via the result's "custom fields". See the example below that demonstrates this workaround.

Workaround

```
// Build a UDF that uses a custom executor to access the address match type from a
// geocode result. The match type value is accessed via the custom field "MATCH_TYPE_LABEL"
val customUdf: UserDefinedFunction = new AddressingBuilder()
    .withResourcesLocation(...)
    .withDataLocations(...)
    .udfBuilder()
    .withOutputFields(..., "customFields['MATCH_TYPE_LABEL'] as MATCH_TYPE_LABEL")
    .forCustomExecutor(new AddressingExecutor {
        override def execute(
            input: RequestInput,
            preferences: Option[Preferences],
            addressing: Addressing): Response = {
```

```

    val response = addressing.geocode(input.requestAddress(), preferences.orNull)
    if(response.getStatus == Status.OK) {
        val firstResult = response.getResults.get(0)
        val matchTypeLabel = firstResult.getExplanation.getAddressMatch.getType.label()
        firstResult.getCustomFields.put("MATCH_TYPE_LABEL", matchTypeLabel)
    }
    response
}
})

```

- (HAD-4323) Some types of queries will cause Hive to evaluate UDFs in the HiveServer2 process space instead of on a data node. The Routing UDFs in particular use a significant amount of memory and can shut down the Hive server due to memory constraints. To process these queries, we recommend increasing the amount of memory available to the HiveServer2 process (for example, by setting HADOOP_HEAPSIZE in hive-env.sh).
- (HAD-2967) A null pointer exception is thrown while running the following constant geocoding queries with Hive on Spark or TEZ:

```
SELECT ReverseGeocode(-76.97921145819674,38.89325106109181,"epsg:4326");
```

```
SELECT Geocode("", "One Second Street", "", "Jersey City", "", "07302", "USA");
```

This is due to an issue in Hive: <https://issues.apache.org/jira/browse/HIVE-16587>

Workaround

```

CREATE table dual (country string);
INSERT into dual values("usa");
SELECT Geocode('','2920 LANGSTON PLACE SOUTHEAST APARTMENT 101',
'', 'WASHINGTON', 'DC', '20020', a.country) from dual a;

```

System Requirements and Dependencies

This product is verified on the following Hadoop distributions:

- Cloudera 5.16, 6.3, 7.1
- Hortonworks 3.0, 3.1
- EMR 5.30, 6.0

To use the product, you must be familiar with configuring Hadoop in Hortonworks, Cloudera, or EMR, and developing applications for distributed processing. For more information, refer to [Hortonworks](#), [Cloudera](#), or [EMR](#) documentation.

The following additional tools must be available to use certain product features:

for Hive:

- Hive version 1.2.1 or above
- Hive in Spectrum Geocoding for Big Data is not supported on Cloudera 5.16

for Spark and Zeppelin Notebook:

- Java JDK version 1.8 or above
- Hadoop version 2.6.0 or above
- Spark version 2.0 or above
- Zeppelin Notebook is not supported in Cloudera

GGG Memory Requirements

- The amount of memory required depends on the deployment scenario implemented. The basic memory requirement for the Spectrum Global Geocoding SDK is 16 GB RAM. We recommend 32 GB RAM.
- If you are using a large number of datasets (more than 20), review your **minimum heap size** setting. Consider increasing it to at least 8 GB to prevent out of memory exception errors.
- For more information, see the [Spectrum Global Geocoding SDK Developer Guide](#).



1700 District Ave Ste 300
Burlington MA 01803-5231
USA

www.precisely.com

© 2015, 2021 Precisely. All rights reserved.