IBM SPSS Analytic Server
Version 2

User's Guide

Note
Before using this information and the product it supports, read the information in "Notices" on page 35.

Product Information

This edition applies to version 2, release 0, modification 0 of IBM SPSS Analytic Server and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. What is new for users in version 2

Analytic Server console

New layout

The layout has been changed so that the pages are accessed via a home page, rather than accordions.

Data sources

- You can define custom attributes for the data source and view custom attributes created by other applications.
- When creating the metadata for a data source, you can initiate a scan of all data values to determine the category values and range limits. Scanning all data values ensures that the metadata is correct, but can take some time if the data source has many fields and records.
- There is support for more types of data sources.

  File content type

  Support for more kinds of file content type includes additional settings and parser formats. You can also define the parsed order of fields for each file in a data source. When adding a directory to a data source, you can specify rules for selecting files within that directory, or its subdirectories.

  Semi-structured files

  These are files, such as web logs, that do not have as much structure as a delimited text file, but contain data that can be extracted into records and fields through regular expressions.

  Compressed files

  Supported compression formats include Gzip, Deflate, Bz2, Snappy, and IBM CMX. Additionally, sequence files with any of the previously mentioned compression formats are supported.

  Text-based files in different formats

  A single text-based data source can now contain documents in different formats (PDF, Microsoft Word, etc.) for text analytics.

  SPSS Statistics files

  SPSS Statistics files (*.sav, *.zsav) are binary files that contain a data model.

  Splittable binary format files (*.asbf)

  This file type can be output by Analytic Server; it is required when the output includes fields that have list values.

  Sequence files

  Sequence files (*.seq) are text files structured as key/value pairs. They are commonly used as an intermediary format in MapReduce jobs.

  Database content type

  You can define data sources for Greenplum, MySQL, and Sybase IQ, if Analytic Server has been configured to be able to use those data sources.

  HCatalog content type

  You can define data sources for Apache Cassandra, MongoDB, and Oracle NoSQL, if Analytic Server has been configured to be able to use those data sources.

  Geospatial content type

  You can define data sources for geographies using shape files or online map services.

Analytics

Support for new SPSS Modeler functionality

Time series

Added support for processing time series, plus distributed building and scoring of temporal causal models (TCM). See the AS Time Intervals, Streaming TCM, and TCM nodes in SPSS Modeler.

Spatial data

Added support for processing geographic coordinate systems, plus distributed building and scoring of association rules and spatio-temporal prediction (STP) models. See the Reprojection, Assocation Rules, and STP nodes in SPSS Modeler.

Clustering

Added support for distributed building and scoring of two-step cluster models. See the TwoStep-AS node in SPSS Modeler.

Improved support for existing SPSS Modeler functionality

Aggregate

String fields can be aggregated using min, max, and count of non-null values. Approximate order statistics (median, quartiles) are supported for numeric fields on the Optimization tab.

Merge Added support for merge by ranked condition.

Added support for merge by condition and merge by keys with no keys; for example, to produce a global mean.

Ensemble modeling

The algorithm for building ensemble models for Tree, Linear, and Neural Net models is improved to better handle data that are not randomly distributed across uniformly sized blocks.

Chapter 2. Analytic Server Console

Analytic Server provides a thin client interface for managing data sources and projects.

Logging in

1. Enter the URL of the Analytic Server in your browser's address bar. The URL can be obtained from your server administrator.
2. Enter the user name with which to log on to the server.
3. Enter the password associated with the specified user name.

After login, the Console home is displayed.

Navigating the Console

- The header displays the product name, the name of the currently logged in user, and the link to the help system. The name of the currently logged in user is the head of a dropdown list that includes the logout link.
- The content area displays the actions you can take from the Console home.

Data sources

A data source is a collection of records, plus a data model, that define a data set for analysis. The source of records can be a file (delimited text, fixed width text, Excel) on HDFS, a relational database, HCatalog, or geospatial. The data model defines all the metadata (field names, storage, measurement level, and so on) necessary for analyzing the data. Data source owners can grant or restrict access to data sources.

Data source listing

The main Data sources page provides a list of data sources of which the current user is a member.

- Click a data source's name to display its details and edit its properties.
- Type in the search area to filter the listing to display only data sources with the search string in their name.
- Click **New** to create a new data source with the name and content type you specify in the **Add new data source** dialog.
  - See "Naming rules" on page 27 for restrictions on the names you can give to data sources.
  - The available content types are File, Database, HCatalog, and Geospatial.

    Note: The HCatalog option is only available if Analytic Server has been configured to work with those data sources.

    Note: The content type cannot be edited once selected.
- Click **Delete** to remove the data source. This action leaves all files associated with the data source intact.
- Click **Refresh** to update the listing.
- The Actions dropdown list performs the selected action.
  1. Select **Export** to create an archive of the data source and save it to the local file system. The archive includes any files that were added to the data source in **Projects** mode or **Data source** mode.
  2. Select **Import** to import an archive created by the Export action.
  3. Select **Duplicate** to create a copy of the data source.

Individual data source details

The content area is divided into several sections, which can depend on the content type of the data source.

Details

These settings are common to all content types.

Name An editable text field that shows the name of the data source.

Display name

An editable text field that shows the name of the data source as displayed in other applications. If this is blank, the Name is used as the display name.

Description

An editable text field to provide explanatory text about the data source.

Is public

A check box that indicates whether anyone can see the data source (checked) or if users and groups must be explicitly added as members (cleared).

Custom attributes

Applications can attach properties to data sources, such as whether the data source is temporary, through the use of custom attributes. These attributes are exposed in the Analytic Server console to provide further insight into how applications use the data source.

Click **Save** to keep the current state of the settings.

Sharing

These settings common to all content types.

You can share ownership of a data source by adding users and groups as authors.

- Typing in the text box filters on users and groups with the search string in their name. Click **Add member** to add them to the list of authors.
- To remove an author, select a user or group in the member list and click **Remove member**.

Note: Administrators have read and write access to every data source, regardless of whether they are specifically listed as a member.

File Input

Settings that are specific to defining data sources with file content type.

File Viewer

Shows available files for inclusion in the data source. Select **Projects** mode to view files within the Analytic Server project structure, **Data source** to view files stored within a data source, or **File system** to view the file system (typically HDFS). You can browse either folder structure, but HDFS is not editable at all, and in **Projects** mode, you cannot add files, create folders, or delete items at the root level, but only within defined projects. To create, edit, or delete a project, use Projects.

- Click **Upload** to upload a file to the current data source or project/subfolder. You can browse for and select multiple files in a single directory.
- Click **New folder** to create a new folder under the current folder, with the name you specify in the New Folder Name dialog.
- Click **Download** to download the selected files to the local file system.
- Click **Delete** to remove the selected files/folders.

Files included in data source definition

Use the move button to add selected files and folders to, or remove them from, the data source. For each selected file or folder in the data source, click Settings to define the specifications for reading the file.

When multiple files are included in a data source, they must share a common metadata; that is, each file must have the same number of fields, the fields must be parsed in the same order in each file, and each field must have the same storage across all files. Mismatches between files can cause the console to fail to create the Preview and Metadata, or otherwise valid values to be parsed as invalid (null) when Analytic Server reads the file.

Database Selections

Specify the connection parameters for the database that contains the record content.

Database

Select the type of database to connect to. Choose from: DB2, Greenplum, MySQL, Netezza, Oracle, SQL Server, Sybase IQ, or TeraData. If the type you are seeking is not listed, ask your server administrator to configure Analytic Server with the appropriate JDBC driver.

Server address

Enter the URL of the server that hosts the database.

Server port

The port number that the database listens on.

Database name

The name of the database you want to connect to.

Username

If the database is password-protected, enter your user name.

Password

If the database is password-protected, enter your password.

Table name

Enter the name of a table from the database that you want to use.

Maximum concurrent reads

Enter the limit on the number of parallel queries that can be sent from Analytic Server to the database to read from the table specified in the data source.

HCatalog Selections

Specify the parameters for accessing data that are managed under Apache HCatalog.

Database

The name of the HCatalog database.

Table name

Enter the name of a table from the database that you want to use.

Filter The partition filter for the table, if the table was created as partitioned table. HCatalog filtering is supported only on Hive partition keys of type string.

Note: The !=, <>, and LIKE operators do not appear to work in certain Hadoop distributions. This is a compatibility issue between HCatalog and those distributions.

HCatalog Field Mappings

Displays the mapping of an element in HCatalog to a field in the data source. Click Edit to modify the field mappings.

Note: After creating an HCatalog based data source that exposes data from a Hive table, you may find that when the Hive table is formed from a large number of data files, there is a substantial delay incurred each time Analytic Server starts to read data from the data source. If you notice such delays, rebuild the Hive table using a smaller number of larger data files, and reduce the number of files to 400 or fewer.

Geospatial Selections

Specify the parameters for accessing geographic data.

Geospatial type

The geographic data can come from an online map service, or a shape file.

If you are using a map service, specify the URL of the service and select the map layer you want to use.

If you are using a shape file, select or upload the shape file. Note that a shape file is actually a set of files with a common filename, stored in the same directory. Select the file with the SHP suffix. Analytic Server will look for and use the other files. Two additional files with the SHX and DBF suffixes must always be present; depending on the shape file, a number of additional files may also be present.

Preview and Metadata

After you specify the settings for the data source, click Preview and Metadata to check and confirm the data source specifications.

Output

Data sources with file or database content type can be appended by output from streams that are run on Analytic Server. Select **Make writeable** to enable appending and:

- For data sources with database content type, choose an output database table where the output data are written.
- For data sources with files content type:
  1. Choose an output folder where the new files are written.

     Tip: Use a separate folder for each data source so it's easier to keep track of the associations between files and data sources.
  2. Select a file format; either **CSV** (comma separated variable) or **Splittable binary format**.
  3. Optionally select **Make sequence file**. This is useful if you want to create splittable compressed files that are usable in downstream MapReduce jobs.
  4. Select **Newlines can be escaped** if your output is CSV and you have string fields that contain embedded newline or carriage return characters. This will cause each newline to be written as a backslash followed by the letter "n", carriage return as a backslash followed by the letter "r", and backslash as two consecutive backslashes. Such data must be read with the same setting. We strongly suggest using the Splittable binary format when handling string data that contains newline or carriage return characters.
  5. Select a compression format. The list includes all formats that have been configured for use with your installation of Analytic Server.

     Note: Some combinations of compression format and file format result in output that cannot be split, and is therefore unsuitable for further MapReduce processing. Analytic Server produces a warning in the Output section when you make such a selection.

Settings (file data sources)

The Settings dialog allows you to define the specifications for reading file-based data. The settings apply to all selected files, and all files within the selected folders that match the criteria on the **Folder** tab.

Specifying the incorrect parser settings for a file can cause the console to fail to create the Preview and Metadata, or otherwise valid values to be parsed as invalid (null) when Analytic Server reads the file.

Settings tab

The Settings tab allows you to specify the file type and parser settings specific to the file type.

You can define data sources using compressed files for any supported file format. Supported compression formats include Gzip, Deflate, Bz2, Snappy, and IBM CMX.

Delimited file type

Delimited files are free-field text files, whose records contain a constant number of fields but a varied number of characters per field. Delimited files typically have *.csv or *.tab file extensions. See "Delimited file type settings" on page 8 for more information.

Fixed file type

Fixed-field text files are files whose fields are not delimited but start at the same position and are of a fixed length. Fixed-field text files typically have a *.dat file extension. See "Fixed file type settings" on page 9 for more information.

Semi-structured file type

Semi-structured files (such as *.log) are text files that have a predictable structure that can be mapped to fields via regular expressions, but are not as highly structured as delimited files. See "Semi-structured file type settings" on page 9 for more information.

Text Analytics file type

Text Analytics files are documents (such as *.doc, *.pdf, or *.txt) that can be analyzed using SPSS Text Analytics.

Skip empty lines
Specifies whether to ignore empty lines in the extracted text content. Default is **No**.

Line separator
Specifies the string that defines a new line. Default is the new line character "\n".

SPSS Statistics file type

SPSS Statistics files (*.sav, *.zsav) are binary files that contain a data model. No further settings on the Settings tab are needed for this file type.

Splittable binary format file type

Specifies that the file type is a splittable binary format file (*.asbf). This file type can represent all Analytic Server field types (unlike CSV, which cannot represent list fields at all and requires special settings to handle embedded newlines and carriage returns). No further settings on the Settings tab are needed for this file type.

Sequence file type

Sequence files (*.seq) are text files structured as key/value pairs. They are commonly used as an intermediary format in MapReduce jobs.

Excel file type

Specifies that the file type is a Microsoft Excel file (*.xls, *.xlsx). See "Excel file type settings" on page 11 for more information.

Delimited file type settings:

You can specify the following settings for delimited file types.

Character set encoding
The character encoding of the file. Select or specify Java charset name such as "UTF-8", "ISO-8859-2", "GB18030". The default is **UTF-8**.

Field delimiters
One or more characters marking field boundaries. Each character is taken as an independent delimiter. For example, if you select **Comma** and **Tab** (or select **Other** and type `,\t`), it means that either a comma or a tab marks field boundaries. If control characters delimit fields, the characters specified here are treated as delimiters in addition to control characters. Default is "," if control characters do not delimit fields; otherwise the default is the empty string.

Control characters delimit fields
Sets whether ASCII control characters, except LF and CR, are treated as field delimiters. Defaults to **No**.

First row contains field names
Sets whether to use the first row to determine the field names. Defaults to **No**.

Number of initial characters to skip
The number of characters at the beginning of the file to be skipped. A non-negative integer. Default is 0.

Merge white space
Sets whether to treat multiple adjacent occurrences of space and/or tab as a single field delimiter. Has no effect if neither space nor tab is a field delimiter. Default is **Yes**.

End-of-line comment characters
One or more characters that mark end-of-line comments. The character and everything following it on the record are ignored. Each character is taken as an independent comment marker. For example, "/*" means either a slash or an asterisk starts a comment. It is not possible to define multi-character comment markers like "//". The empty string signals that no comment characters are defined. If defined, comment characters are checked for before quotes are processed or initial characters to skip are skipped. Default is the empty string.

Invalid characters
Determines how invalid characters (byte sequences that do not correspond to characters in the encoding) are to be handled.

Discard
Discard invalid byte sequences.

Replace with
Replace each invalid byte sequence with the given single character.

Single quotes
Specifies handling of single quotes (apostrophes). Default is **Keep**.

Keep Single quotes have no special meaning and are treated as any other character.

Drop Single quotes are deleted unless quoted

Pair Single quotes are treated as quote characters and characters between pairs of single quotes lose any special meaning (they are considered quoted). Whether single quotes themselves can occur inside single-quoted strings is determined by the setting **Quotes can be quoted by doubling**.

Double quotation marks
Specifies handling of double quotation marks. Default is **Pair**.

Keep Double quotation marks have no special meaning and are treated as any other character.

Drop Double quotation marks are deleted unless quoted

Pair Double quotation marks are treated as quote characters and characters between pairs of double quotation marks lose any special meaning (they are considered quoted). Whether double quotation marks themselves can occur inside double-quoted strings is determined by the setting **Quotes can be quoted by doubling**.

Quotes can be quoted by doubling
Indicates whether double quotation marks can be represented in double-quoted strings and single quotes can be represented in single-quoted strings when set to **Pair**. If **Yes**, double quotation marks are escaped inside double-quoted strings by doubling and single quotes are escaped inside single-quoted strings by doubling. If **No**, there is no way to quote a double quote inside a double-quoted string or a single quote inside a single-quoted string. Default is **Yes**.

Newlines can be escaped

Indicates whether the parser interprets a backslash followed by the letter "n", the letter "r" or another backslash as a newline, carriage return or backslash character, respectively. If newlines are not escaped, those character sequences are read literally as a backslash followed by the letter "n", and so on. Default is **No**.

Fixed file type settings:

You can specify the following settings for fixed file types.

Character set encoding
The character encoding of the file. Select or specify Java charset name such as "UTF-8", "ISO-8859-2", "GB18030". The default is **UTF-8**.

Invalid characters
Determines how invalid characters (byte sequences that do not correspond to characters in the encoding) are to be handled.

Discard
Discard invalid byte sequences.

Replace with
Replace each invalid byte sequence with the given single character.

Record length
Indicates how records are defined. If **Newline delimited**, records are defined (delimited) by newlines, beginning of file, or end of file. If **Specific length**, records are defined by a record length in bytes. Specify a positive value.

Initial records to skip
The number of records at the beginning of the file to be skipped. Specify a non-negative integer. The default value is 0.

Fields This section defines the fields in the file. Click **Add Field** and specify the field name, the column in which the field values start, and the length of the field values. Columns in a file are numbered starting at 0.

Semi-structured file type settings:

Settings for semi-structured files consist of rules for mapping the file's contents to fields.

Rules Table
Individual rules extract information from a record to create a field; together in the rules table, they define all of the fields that can be extracted from each record in a data source.

The rules in the table are applied in order to each record; if all of the rules in the table match the record, then any other rules tables are not needed to process the record, and the next record is processed. If any rule in the table does not match, then all field values extracted by previous

rules in the table are discarded; if there is another rules table, the rules in that table is are applied to the record. If no table matches the record, then the Mismatch rule is applied.

Mismatch
You can choose to **Skip** records that do not match any of the rules tables, or set the value of all fields in the record to **Missing** (null).

Export Rules
You can save the currently visible rules table for reuse. The exported table is saved on the server.

Import Rules
You can import a saved rules table into the currently visible rules table. This overwrites any rules you have defined for that table, so it's best to create a new table and then import a rules table.

Rule Editor

The rule editor allows you to create an extraction rule for a single field.

Anonymous capture group
A field capture rule typically starts to extract data from a record at the position where the previous rule stopped. When there is extraneous information between two fields in a semi-structured data source, it can therefore be useful to define an anonymous capture group that positions the parser where the next field begins. When you select **Anonymous capture group**, the controls for naming and labeling the capture group are disabled, but the rest of the dialog functions normally.

Field name
Enter a name for the field. This is used to define the data source metadata. Field names must be unique within a rules table.

Rule name
Optionally enter a descriptive label for the rule.

Description
Optionally enter a longer description for the rule.

Defining a rule
There are two methods for defining rules.

Use controls for extraction rules
This simplifies the creation of extraction rules.

1. Specify the point to start extracting field data; **Current position** will start where the previous rule stopped, and **Skip until** will start at the beginning of the record and ignore all characters until it reaches the one specified in the text box. Select **Include** if you want the field data to include the character at the start position.
2. Select a field capture group from the **Capture** dropdown.
3. Optionally select the point to stop extracting field data; **Whitespace** will stop when any whitespace characters (such as spaces or tabs) are encountered, and **At character(s)** will stop at the specified string. Select **Include** if you want the field data to include the character at the stop position.

Manually define regexp rules
Select this if you are comfortable writing regular expression syntax. Enter a regular expression into the **Regexp** text box.

Add Field Capture Group
This allows you to save the regular expression for later use. The saved capture group appears on the **Capture** dropdown.

The Rule Editor shows a preview of the data extracted from the first record by this rule, after all previous rules in the rules table have been applied.

Excel file type settings:

You can specify the following settings for Excel files.

Worksheet selection
Selects the Excel worksheet to be used as the data source. Specify either a numeric index (the index of the first worksheet is 0) or the name of the worksheet. The default is to use the first worksheet.

Data range selection for import.
You can import data beginning with the first non-blank row or with an explicit range of cells.

- **Range starts on first non-blank row**. Locates the first non-blank cell and uses this as the upper left corner of the data range.
- Alternatively specify an explicit range of cells by row and column. For example, to specify the Excel range A1:D5, you can enter A1 in the first field and D5 in the second (or alternatively, R1C1 and R5C4). All rows in the specified range are returned, including blank rows.

First row contains field names
Specifies whether the first row of the selected cell range contains the field names. The default is **No**.

Stop reading after encountering blank rows
Specifies whether to stop reading records after more than one blank row is encountered, or to continue reading all data to the end of the worksheet, including blank rows. The default is **No**.

Formats

The Formats tab allows you to define formatting information for the parsed fields.

Field Conversion Settings

Trim white space
Removes white space characters from the beginning and/or end of the string fields. Defaults to **None**. The following values are supported:

None Does not remove white space characters.

Left Removes white space characters from the beginning of the string.

Right Removes white space characters from the end of the string.

Both Removes white space characters from the beginning and end of the string.

Locale Defines a locale. Defaults to the server locale. The locale string should be specified as: <language>[_country[_variant]], where:

language
A valid, lower-case, two-letter code as defined by ISO-639.

country
A valid, upper-case, two-letter code as defined by ISO-3166.

variant
A vendor or browser-specific code.

Decimal separator
Sets the character used as the decimal sign. Defaults to the locale-specific setting.

Grouping symbols
Sets whether or not the locale-specific character used for the thousands separator should be used.

Default date format
Defines a default date format. All format patterns defined by the unicode locale data markup language (LDML) specification are supported.

Default time format
Defines a default time format.

Default timestamp
Defines a default timestamp format.

Default time zone
Sets the timezone. Defaults to UTC. The setting applies to the time and timestamp fields which do not have explicitly specified timezone.

Field Overrides

This section allows you to assign formatting instructions to individual fields. Select a field from the data model, or type in a field name, and click **Add** to add it to the list of fields with individual instructions. Click **Remove** to remove it from the list. For a selected field in the list, you can set the following properties of the field.

Storage
Set the storage of the field.

Decimal separator
For fields with Real storage, sets the character used as the decimal sign. Defaults to the locale-specific setting.

Grouping symbols
For fields with Integer or Real storage, sets whether or not the locale-specific character used for the thousands separator should be used.

Formats
For fields with Date, Time, or Timestamp storage, sets the format. Choose a format from the dropdown list.

Field Order tab

For delimited and Excel file types, the Field Order tab allows you to define the parsed order of fields for the file. This is important when there are multiple files in a data source, because the actual order of fields may be different across the files, but the parsed order of fields must be the same to create a consistent data model.

For fixed and semi-structured file types, the order is defined on the Settings tab.

When there is a single file in the data source, or all files have the same field order, you can use the default **Field order matches data model**. If there are multiple files in the data source, and order of fields in the file do not match, define a **Specific field order** for parsing the file.

1. To add a field to the ordered list, type the field name or select it from the list provided by the data model. You can add all fields in the data model at once by clicking **Add all**. Field names will only be added once to the ordered list.
2. Use the arrow buttons to order the fields as desired.

When **Specific field order** is used, any fields not added to the list are not part of the result set for this file. If there are fields are in the data model that are not listed in this dialog, the values are null in the result set.

Folder tab

When specifying parser settings for a folder, the Folder tab allows you to choose which files in the folder are included in the data source.

Match all files in the selected folder
The data source includes all files in the top level of the folder; files in subfolders are not included.

Match files using a regular expression
The data source includes all files in the top level of the folder that match the specified regular expression; files in subfolders are not included.

Match files using a Unix globbing expression (potentially recursive)
The data source includes all files that match the specified Unix globbing expression; the expression can include files that are in subfolders of the selected folder.

HCatalog Field Mappings

HCatalog Schema
Displays the structure of the specified table. HCatalog can support a highly structured data set. To define an Analytic Server data source on such data, the structure must be flattened into simple rows and columns. Select an element in the schema and click the move button to map it to a field for analysis.

Not all tree nodes can be mapped. For example, an array or map of complex types is considered a "parent" and cannot be directly mapped; each simple element in an HCatalog array or map must be added separately. These nodes can be identified by the label in the tree ending in ...:array:struct, or ...:map:struct.

For example:

- For an array of integers, you can assign a field to a value within the array: bigintarray[45], but not the array itself: bigintarray
- For a map, you can assign a field to a value within the map: datamap["key"], but not the map itself: datamap
- For an array of an array of integers, you can assign a field to a value bigintarrayarray[45][2], but not the array itself, bigintarrayarray[45].

  Therefore, when you assign a field to an array or map element, the definition of the element must include the index or key: bigintarray[index] or bigintmap["key"].

Field Mappings

HCatalog Element
Double-click a cell to edit. You must edit the cell when the HCatalog element is an array or map. With an array, specify the integer corresponding to the member of the array you want to map to a field. With a map, specify a quoted string corresponding to the key you want to map to a field.

Mapping Field
The field as it appears in the Analytic Server data source. Double-click a cell to edit. Duplicate values in the Mapping Field column are not allowed and result in an error.

Storage
The storage of the field. Storage is derived from HCatalog and cannot be edited.

Note: When you click Preview and Metadata to finalize an HCatalog data source, there are no editing options.

Raw Data
Displays the records as they are stored in HCatalog; this can help you determine how to map the HCatalog schema to fields.

Note: Any filtering specified in the HCatalog Selections is applied to the view of the raw data.

Enabling HCatalog data sources

Analytic Server provides support for HCatalog data sources. This section describes how to enable various underlying NoSQL databases.

Apache Accumulo

Analytic Server provides support for HCatalog data sources that have underlying content in Apache Accumulo.

The Apache Accumulo distributed key/value store is a data storage and retrieval system, based on Google's BigTable design and is built on top of Apache Hadoop, Zookeeper, and Thrift. Apache Accumulo features a few novel improvements on the BigTable design in the form of cell-based access control and a server-side programming mechanism that can modify key/value pairs at various points in the data management process.

To create an external Apache Accumulo table in Hive use the following syntax:

```
set accumulo.instance.id=<instance_name>;
set accumulo.user.name=<user_name>;
set accumulo.user.pass=<user_password>;
set accumulo.zookeepers=<zookeeper_host_port>;

CREATE EXTERNAL TABLE <hive_table_name>(<table_column_specifications>)
STORED BY 'com.ibm.spss.hcatalog.AccumuloStorageHandler'
WITH SERDEPROPERTIES (
'accumulo.columns.mapping' = '<family_and_qualifier_mappings>',
'accumulo.table.name' = '<Accumulo_table_name>')
TBLPROPERTIES (
    "accumulo.instance.id"="<instance_name>",
    "accumulo.zookeepers"="<zookeeper_host_port>"
);
```

For example:

```
set accumulo.instance.id=<id>;
set accumulo.user.name=admin;
set accumulo.user.pass=test;
set accumulo.zookeepers=<host>:<port>;

CREATE EXTERNAL TABLE acc_drug1n(rowid STRING,age STRING,sex STRING,bp STRING,
    cholesterol STRING,na STRING,k STRING,drug STRING)
STORED BY 'com.ibm.spss.hcatalog.AccumuloStorageHandler'
WITH SERDEPROPERTIES (
'accumulo.columns.mapping' = 'rowID,drug|age,drug|sex,drug|bp,drug|cholesterol,
    drug|na,drug|k,drug|drug',
'accumulo.table.name' = 'drug1n')
TBLPROPERTIES (
    "accumulo.instance.id"="<id>",
    "accumulo.zookeepers"="<host>:<port>"
);
```

Note: The Accumulo user name and password for the given Accumulo table should match the user name and password of the authenticated Analytic Server user.

Apache Cassandra

Analytic Server provides support for HCatalog data sources that have underlying content in Apache Cassandra.

Cassandra provides a structured key-value store. Keys map to multiple values, which are grouped into column families. The column families are fixed when a database is created, but columns can be added to a family at any time. Furthermore, columns are added only to specified keys, so different keys can have different numbers of columns in any given family. The values from a column family for each key are stored together.

There are two ways to define Cassandra tables: using the legacy Cassandra command line interface (cassandra-cli) and the new CQL shell (csqlsh).

Use the following syntax to create an external Apache Cassandra table in Hive if the table was created using legacy CLI.

```
CREATE EXTERNAL TABLE <hive_table_name>  (<column specifications>)
STORED BY 'com.ibm.spss.hcatalog.CassandraStorageHandler'
WITH SERDEPROPERTIES("cassandra.cf.name" = "<cassandra_column_family>",
"cassandra.host"="<cassandra_host>","cassandra.port" = "<cassandra_port>")
TBLPROPERTIES ("cassandra.ks.name" = "<cassandra_keyspace>");
```

For example, for the following CLI table definition:

```
create keyspace test
with placement_strategy = 'org.apache.cassandra.locator.SimpleStrategy'
and strategy_options = [{replication_factor:1}];

create column family users with comparator = UTF8Type;

update column family users with
        column_metadata =
        [
        {column_name: first, validation_class: UTF8Type},
        {column_name: last, validation_class: UTF8Type},
        {column_name: age, validation_class: UTF8Type, index_type: KEYS}
        ];

assume users keys as utf8;

set users['jsmith']['first'] = 'John';
set users['jsmith']['last'] = 'Smith';
set users['jsmith']['age'] = '38';
set users['jdoe']['first'] = 'John';
set users['jdoe']['last'] = 'Dow';
set users['jdoe']['age'] = '42';

get users['jdoe'];
```

... the Hive table DDL will look like this:

```
CREATE EXTERNAL TABLE cassandra_users  (key string, first string, last string, age string)
STORED BY 'com.ibm.spss.hcatalog.CassandraStorageHandler'
WITH SERDEPROPERTIES("cassandra.cf.name" = "users",
"cassandra.host"="<cassandra_host>","cassandra.port" = "9160")
TBLPROPERTIES ("cassandra.ks.name" = "test");
```

Use the following syntax to create an external Apache Cassandra table in Hive if the table was created using CQL.

```
CREATE EXTERNAL TABLE <hive_table_name>  (<column specifications>)
STORED BY 'com.ibm.spss.hcatalog.CassandraCqlStorageHandler'
WITH SERDEPROPERTIES("cassandra.cf.name" = "<cassandra_column_family>",
"cassandra.host"="<cassandra_host>","cassandra.port" = "<cassandra_port>")
TBLPROPERTIES ("cassandra.ks.name" = "<cassandra_keyspace>");
```

For example, for the following CQL3 table definition:

```
CREATE KEYSPACE TEST WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 2 };
USE TEST;

CREATE TABLE bankloan_10(
  row int,
  age int,
  ed int,
  employ int,
  address int,
  income int,
  debtinc double,
  creddebt double,
  othdebt double,
```

```
  default int,
  PRIMARY KEY(row)
);

INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (1,41,3,17,12,176,9.3,11.359392,5.008608,1);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (2,27,1,10,6,31,17.3,1.362202,4.000798,0);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (3,40,1,15,14,55,5.5,0.856075,2.168925,0);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (4,41,1,15,14,120,2.9,2.65872,0.82128,0);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (5,24,2,2,0,28,17.3,1.787436,3.056564,1);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (6,41,2,5,5,25,10.2,0.3927,2.1573,0);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (7,39,1,20,9,67,30.6,3.833874,16.668126,0);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (8,43,1,12,11,38,3.6,0.128592,1.239408,0);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (9,24,1,3,4,19,24.4,1.358348,3.277652,1);
INSERT INTO bankloan_10 (row, age,ed,employ,address,income,debtinc,creddebt,othdebt,default)
   VALUES (10,36,1,0,13,25,19.7,2.7777,2.1473,0);
```

... the Hive table DDL is as follows:

```
CREATE EXTERNAL TABLE cassandra_bankloan_10 (row int, age int,ed int,employ int,address int,
   income int,debtinc double,creddebt double,othdebt double,default int)
STORED BY 'com.ibm.spss.hcatalog.CassandraCqlStorageHandler'
WITH SERDEPROPERTIES("cassandra.cf.name" = "bankloan_10","cassandra.host"="<cassandra_host>",
   "cassandra.port" = "9160")
TBLPROPERTIES ("cassandra.ks.name" = "test");
```

Apache HBase

Analytic Server provides support for HCatalog data sources that have underlying content in Apache HBase.

Apache HBase is an open-source, distributed, versioned, column-oriented store on top of Hadoop and HDFS.

To create an external HBase table in Hive use the following syntax:

```
CREATE EXTERNAL TABLE <tablename>(<table_column_specifications>)
STORED BY 'com.ibm.spss.hcatalog.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" = "<column_mapping_spec>")
TBLPROPERTIES("hbase.table.name" = "<hbase_table_name>")
```

For example:

```
CREATE EXTERNAL TABLE hbase_drug1n(rowid STRING,age STRING,sex STRING,bp STRING,
    cholesterol STRING,na STRING,k STRING,drug STRING)
STORED BY 'com.ibm.spss.hcatalog.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,drug:age,drug:sex,drug:bp,
    drug:cholesterol,drug:na,drug:k,drug:drug")
TBLPROPERTIES("hbase.table.name" = "drug1n");
```

Note: For information on how to create an HBase table, see the Apache HBase Reference Guide (http://hbase.apache.org/book.html).

Note: It is a good practice to preface the database name to indicate the type of database. For example, name a database HB_drug1n to indicate an HBase database, or ACC_drug1n to indicate an Accumulo database. This will help with the selection of the HCatalog file when in the Analytic Server console.

MongoDB

Analytic Server provides support for HCatalog data sources that have underlying content in MongoDB.

MongoDB is an open source document database, and the leading NoSQL database written in C++. The database stores JSON-style documents with dynamic schemas.

To create an external MongoDB table in Hive use the following syntax:

```
create external table <hive_table_name>(<column specifications>)
stored by "com.ibm.spss.hcatalog.MongoDBStorageHandler"
with serdeproperties ( "mongo.column.mapping" = "<MongoDB to Hive mapping>" )
tblproperties ( "mongo.uri" = "'mongodb://<host>:<port>/<database>.<collection>" );
```

For example:

```
create external table mongo_bankloan(age bigint,ed bigint,employ bigint, address bigint,income bigint,
    debtinc double, creddebt double,othdebt double,default bigint)
STORED BY 'com.ibm.spss.hcatalog.MongoDBStorageHandler'
with serdeproperties ( 'mongo.column.mapping' = '{"age":"age","ed":"ed","employ":"employ","address":"address",
    "income":"income","debtinc":"debtinc","creddebt":"creddebt","othdebt":"othdebt","default":"default"}' )
tblproperties ('mongo.uri'='mongodb://9.48.11.162:27017/test.bankloan');
```

Oracle NoSQL

Analytic Server provides support for HCatalog data sources that have underlying content in Oracle NoSQL.

The Oracle NoSQL Database is a distributed key-value database. Data is stored as key-value pairs, which are written to particular storage nodes, based on the hashed value of the primary key. Storage nodes are replicated to ensure high availability. Customer applications are written using Java/C API to read and write data.

SerDe and table parameters

The Oracle NoSQL storage handler supports the following parameters.

SERDEPROPERTIES parameters

kv.major.keys.mapping
Comma separated list of the major keys. Required

kv.minor.keys.mapping
Comma separated list of the minor keys. Optional

kv.parent.key
Specifies the parent key whose "child" key-value pairs are to be returned by the query. The major key path must be a partial path and the minor key path must be empty. Optional.

kv.avro.json.key
The name of the minor key that is used to hold the value defined with the Avro schema. If the minor key is not defined, which is typically the case, defaults to "value". If the parameter is not defined the value will be returned as a JSON string. Optional.

kv.avro.json.keys.mapping.column
Defines the name of the Hive column for the major/minor key-value pairs. The Hive column should have map<string,string> type. Optional.

TABLEPROPERTIES parameters

kv.host.port
The IP address and the port number of the Oracle NoSQL database. Required

kv.name
The name of the Oracle NoSQL key-value store. Required.

Example: simple Avro schema

The data layout is modeled using Apache Avro serialization framework. To follow this approach you create an Avro schema; for example:

```
{"type":"record",
 "name":"DrugSchema",
 "namespace":"avro",
 "fields":[
        {"name":"id","type":"string","default":""},
        {"name":"age","type":"string","default":""},
        {"name":"sex","type":"string","default":""},
        {"name":"bp","type":"string","default":""},
        {"name":"drug","type":"string","default":""}
]}
```

This schema should be registered with the Oracle NoSQL database and the populated data should include a reference to the schema as shown below.

```
put -key /drugstore_avro/1 -value
   "{\"id\":\"1\",\"age\":\"23\",\"sex\":\"F\",\"bp\":\"HIGH\",\"drug\":\"drugY\"}"
   -json avro.DrugSchema
put -key /drugstore_avro/2 -value
   "{\"id\":\"2\",\"age\":\"47\",\"sex\":\"M\",\"bp\":\"LOW\",\"drug\":\"drugC\"}"
   -json avro.DrugSchema
put -key /drugstore_avro/3 -value
   "{\"id\":\"3\",\"age\":\"47\",\"sex\":\"M\",\"bp\":\"LOW\",\"drug\":\"drugC\"}"
   -json avro.DrugSchema
put -key /drugstore_avro/4 -value
   "{\"id\":\"4\",\"age\":\"28\",\"sex\":\"F\",\"bp\":\"NORMAL\",\"drug\":\"drugX\"}"
   -json avro.DrugSchema
put -key /drugstore_avro/5 -value
   "{\"id\":\"5\",\"age\":\"61\",\"sex\":\"F\",\"bp\":\"LOW\",\"drug\":\"drugY\"}"
   -json avro.DrugSchema
```

In order to expose the data in Hive, create an external table and specify the additional property **kv.avro.json.key** in the SERDEPROPERTIES section. The value of the property should be the name of the minor key or the predefined name **value** if the minor key is not defined.

```
CREATE EXTERNAL TABLE oracle_json(id string, age string, sex string, bp string, drug string)
  STORED BY 'com.ibm.spss.hcatalog.OracleKVStorageHandler'
  WITH SERDEPROPERTIES ("kv.major.keys.mapping" = "drugstore_avro,keyid",
        "kv.parent.key"="/drugstore_avro","kv.avro.json.key" = "value")
  TBLPROPERTIES ("kv.host.port" = "<hostname>:5000", "kv.name" = "kvstore");
```

Running select * from oracle_json produces the following results.

```
select * from oracle_json;

1 23 F HIGH drugY
5 61 F LOW drugY
3 47 M LOW drugC
2 47 M LOW drugC
4 28 F NORMAL drugX
```

The table oracle_json can be used in the Analytic Server console to create an Oracle NoSQL data source.

Example: complex keys

Now consider the following Avro schema.

```
{ "type": "record",
  "name": "DrugSchema",
  "namespace":"avro",
  "fields": [
      {"name": "age", "type": "string", "default": ""}, // age
      {"name": "bp", "type":"string","default":""},     // blood pressure
      {"name": "drug", "type": "int", "default": ""},    // drug administered
]}
```

Also assume that the key is modeled as follows:

```
/u/<sex (M/F)>/<patient ID>
```

and populate the data store using these commands:

```
put -key /u/F/1 -value
  "{\"age\":\"23\",\"bp\":\"HIGH\",\"drug\":\"drugY\"}" -json avro.DrugSchema
put -key /u/M/2 -value
  "{\"age\":\"47\",\"bp\":\"LOW\",\"drug\":\"drugC\"}" -json avro.DrugSchema
put -key /u/M/3 -value
  "{\"age\":\"47\",\"bp\":\"LOW\",\"drug\":\"drugC\"}" -json avro.DrugSchema
put -key /u/F/4 -value
  "{\"age\":\"28\",\"bp\":\"NORMAL\",\"drug\":\"drugX\"}" -json avro.DrugSchema
put -key /u/F/5 -value
  "{\"age\":\"61\",\"bp\":\"LOW\",\"drug\":\"drugY\"}" -json avro.DrugSchema
```

To preserve the information about the gender and user ID from the major keys the table should be created with an additional SERDEPROPERTIES parameter **kv.avro.json.keys.mapping.column**. The value of the parameter should be the name of the Hive column of type map<string,string> . The keys in the map will be the names of the record keys specified in the **kv.*.keys.mapping** properties and the values will be the actual key values. The table creation DDL is shown below:

```
CREATE EXTERNAL TABLE oracle_user(keys map<string,string>, age string, bp string, drug string)
  STORED BY 'com.ibm.spss.hcatalog.OracleKVStorageHandler'
  WITH SERDEPROPERTIES ("kv.major.keys.mapping" = "DrugSchema,sex,patientid",
      "kv.parent.key" = "/u",
      "kv.avro.json.key" = "value",
      "kv.avro.json.keys.mapping.column" = "keys")
  TBLPROPERTIES ("kv.host.port" = "<hostname>:5000", "kv.name" = "kvstore");
```

Running select * from oracle_user will produce the following results :

```
select * from
        oracle_user; {"user":"u","gender":"m"," userid":"125"} joe smith 77 13
      {"user":"u","gender":"m"," userid":"129"} jeff smith 67 27
      {"user":"u","gender":"m"," userid":"127"} jim smith 78 11
      {"user":"u","gender":"f"," userid":"131"} jen schmitt 70 20
      {"user":"u","gender":"m"," userid":"130"} jed schmidt 60 31
      {"user":"u","gender":"f"," userid":"128"} jan smythe 79 10
      {"user":"u","gender":"f"," userid":"126"} jess smith 76 12
```

The oracle_user table can be used in the Analytic Server console to create an Oracle NoSQL data source. The sex and patientid keys as well as the column names from the Avro schema can be used to define corresponding fields for the data source.

Range scans

Analytic Server supports range scans based on the parent prefix for the major keys as well as subranges to further restrict the range under the parent key.

The parent key specifies the prefix for the "child" key-value pairs to be returned. An empty prefix results in fetching all keys in the store. If the prefix is not empty, the major key path must be a partial path and the minor key path must be empty. The parent key is stored as a **com.ibm.spss.ae.hcatalog.range.parent** data source attribute.

The subrange further restricts the range under the parent key to the major path components in the subrange. The subrange start key is stored as **com.ibm.spss.ae.hcatalog.range.start** and the subrange end key is stored as **com.ibm.spss.ae.hcatalog.range.end**. The start key should be lexicographically less than or equal to the end key. The subrange parameters are optional.

XML data sources

Analytic Server provides support for XML data through HCatalog.

Example

1. Map the XML schema to Hive data types through the Hive Data Definition Language (DDL), according to the following rules.

```
CREATE [EXTERNAL] TABLE <table_name> (<column_specifications>)
ROW FORMAT SERDE "com.ibm.spss.hive.serde2.xml.XmlSerDe"
WITH SERDEPROPERTIES (
    ["xml.processor.class"="<xml_processor_class_name>",]
    "column.xpath.<column_name>"="<xpath_query>",
     ...
    ["xml.map.specification.<element_name>"="<map_specification>"
     ...
    ]
)
STORED AS
    INPUTFORMAT "com.ibm.spss.hive.serde2.xml.XmlInputFormat"
    OUTPUTFORMAT "org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat"
[LOCATION "<data_location>"]
TBLPROPERTIES (
    "xmlinput.start"="<start_tag ",
    "xmlinput.end"="<end_tag>"
);
```

Note: If your XML files are compressed with Bz2 compression, then `INPUTFORMAT` should be set to `com.ibm.spss.hive.serde2.xml.SplittableXmlInputFormat`. If they are compressed with CMX compression, it should be set to `com.ibm.spss.hive.serde2.xml.CmxXmlInputFormat`.

For example, the following XML...

```
<records>
  <record customer_id="0000-JTALA">
    <demographics>
      <gender>F</gender>
      <agecat>1</agecat>
      <edcat>1</edcat>
      <jobcat>2</jobcat>
      <empcat>2</empcat>
      <retire>0</retire>
      <jobsat>1</jobsat>
      <marital>1</marital>
      <spousedcat>1</spousedcat>
      <residecat>4</residecat>
      <homeown>0</homeown>
      <hometype>2</hometype>
      <addresscat>2</addresscat>
    </demographics>
    <financial>
      <income>18</income>
      <creddebt>1.003392</creddebt>
      <othdebt>2.740608</othdebt>
      <default>0</default>
    </financial>
  </record>
</records>
```

...would be represented by the following Hive DDL.

```
CREATE TABLE xml_bank(customer_id STRING, demographics map<string,string>, financial map<string,string>)
ROW FORMAT SERDE 'com.ibm.spss.hive.serde2.xml.XmlSerDe'
WITH SERDEPROPERTIES (
    "column.xpath.customer_id"="/record/@customer_id",
    "column.xpath.demographics"="/record/demographics/*",
    "column.xpath.financial"="/record/financial/*"
)
STORED AS
    INPUTFORMAT 'com.ibm.spss.hive.serde2.xml.XmlInputFormat'
    OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
TBLPROPERTIES (
    "xmlinput.start"="<record customer",
    "xmlinput.end"="</record>"
);
```

See "XML to Hive Data Types Mapping" for more information.

2. Create an Analytic Server data source with HCatalog content type in the Analytic Server Console.

Limitations

- Only the XPath 1.0 specification is currently supported.
- The local part of the qualified names for the elements and attributes are used when handling Hive field names. The namespace prefixes are ignored.

XML to Hive Data Types Mapping: The data modeled in XML can be transformed to the Hive data types using the conventions documented below.

Structures

The XML element can be directly mapped to the Hive structure type so that all the attributes become the data members. The content of the element becomes an additional member of primitive or complex type.

XML data

```
<result name="ID_DATUM">03.06.2009</result>
```

Hive DDL and raw data

```
struct<name:string,result:string>
{"name":"ID_DATUM", "result":"0.3.06.2009"}
```

Arrays

The XML sequences of elements can be represented as Hive arrays of primitive or complex type. The following example shows how the user can define an array of strings using content of the XML <result> element.

XML data

```
<result>03.06.2009</result>
<result>03.06.2010</result>
<result>03.06.2011</result>
```

Hive DDL and raw data

```
result array<string>
{"result":["03.06.2009","03.06.2010",...]}
```

Maps

The XML schema does not provide native support for maps. There are three common approaches to modeling maps in XML. To accommodate the different approaches we use the following syntax:

```
"xml.map.specification.<element_name>"="<key>-><value>"
```

where

element_name
The name of the XML element to be considered as a map entry

key The map entry key XML node

value The map entry value XML node

The map specification for the given XML element should be defined under the SERDEPROPERTIES section in the Hive table creation DDL. The keys and values can be defined using the following syntax:

@attribute
The @attribute specification allows the user to use the value of the attribute as a key or value of the map.

element
The element name can be used as a key or value.

#content
The content of the element can be used as a key or value. As the map keys can only be of primitive type the complex content will be converted to string.

The approaches to representing maps in XML, and their corresponding Hive DDL and raw data, is as follows.

Element name to content
The name of the element is used as a key and the content as a value. This is one of the common techniques and is used by default when mapping XML to Hive map types. The obvious limitation with this approach is that the map key can be only of type string.

XML data

```
<entry1>value1</entry1>
<entry2>value2</entry2>
<entry3>value3</entry3>
```

Mapping, Hive DDL, and raw data

In this case you do not need to specify a mapping because the name of the element is used as a key and the content as a value by default.

```
result map<string,string>
```

```
{"result":{"entry1": "value1", "entry2": "value2", "entry3": "value3"}}
```

Attribute to Element Content
Use an attribute value as a key and the element content as a value.

XML data

```
<entry name="key1">value1</entry>
<entry name="key2">value2</entry>
<entry name="key3">value3</entry>
```

Mapping, Hive DDL, and raw data

```
"xml.map.specification.entry"="@name->#content"
```

```
result map<string,string>
```

```
{"result":{"key1": "value1", "key2": "value2", "key3": "value3"}}
```

Attribute to Attribute

XML data

```
<entry name="key1" value="value1"/>
<entry name="key2" value="value2"/>
<entry name="key3" value="value3"/>
```

Mapping, Hive DDL, and raw data

```
"xml.map.specification.entry"="@name->@value"
```

```
result map<string,string>
```

```
{"result":{"key1": "value1", "key2": "value2", "key3": "value3"}}
```

Complex Content

Complex content being used as a primitive type will be converted to a valid XML string by adding a root element called <string>. Consider the following XML:

```
<dataset>
 <value>10</value>
 <value>20</value>
 <value>30</value>
</dataset>
```

The XPath expression /dataset/* will result in a number of <value> XML nodes being returned. If the target field is of primitive type the implementation will transform the result of the query to the valid XML by adding the <string> root node.

```
<string>
 <value>10</value>
 <value>20</value>
 <value>30</value>
</string>
```

Note: The implementation will not add a root element <string> If the result of the query is a single XML element.

Text Content

The whitespace-only text content of an XML element is ignored.

Preview and Metadata (data sources)

Clicking **Preview and Metadata** displays a sample of records and the data model for the data source. Here you have a chance to review the basic metadata information.

Preview

The Preview tab shows a small sample of records and their field values.

Edit

The Edit tab displays the basic field metadata. For data sources with Files content type, the data model is generated from a small sample of records, and you can manually edit the field metadata on this tab. For data sources with HCatalog content type, the data model is generated based upon the HCatalog Field Mappings, and you cannot edit the field storage on this tab.

Field Double-click on the field name to edit it.

Measurement

This is the measurement level, used to describe characteristics of the data in a given field.

Role Used to tell modeling nodes whether fields will be Input (predictor fields) or Target (predicted fields) for a machine-learning process. Both and None are also available roles, along with Partition, which indicates a field used to partition records into separate samples for training, testing, and validation. The value Split specifies that separate models will be built for each possible value of the field. Frequency specifies that a field values should be used as a frequency weight for each record. Record ID is used to identify a record in the output.

Storage

Storage describes the way data are stored in a field. For example, a field with values of 1 and 0 stores integer data. This is distinct from the measurement level, which describes the

usage of the data, and does not affect storage. For example, you may want to set the measurement level for an integer field with values of 1 and 0 to Flag. This usually indicates that 1 = True and 0 = False.

Values
Shows the individual values for fields with categorical measurement, or the range of values for fields with continuous measurement.

Structure
Indicates whether records in the field contain a single value (Primitive) or a list of values.

Depth Indicates the depth of a list; 0 is a list of primitives, 1 is a list of lists, and so on.

Scan all Data Values
This allows you to initiate and cancel the scan of the data source data values to determine the category values and range limits. If a scan is in progress, click the button to **Cancel Data Scan**. Scanning all data values ensures that the metadata is correct, but can take some time if the data source has many fields and records.

Projects

Projects are workspaces for storing inputs and accessing outputs of jobs. They provide the top-level organizational structure for containing files and folders. Projects can be shared with individual users and groups.

Project listing

The main Projects page provides a list of projects of which the current user is a member.

- Click a project's name to display its details and edit its properties.
- Type in the search area to filter the listing to display only projects with the search string in their name.
- Click **New** to create a new project with the name you specify in the **Add new project** dialog. See "Naming rules" on page 27 for restrictions on the names you can give to projects.
- Click **Delete** to remove the selected project(s). This action removes the project and deletes all data associated with the project from HDFS.
- Click **Refresh** to update the listing.

Individual project details

The content area is divided into **Details**, **Sharing**, **Files**, and **Versions** collapsible sections.

Details

Name An editable text field that shows the name of the project.

Display name
An editable text field that shows the name of the project as displayed in other applications. If this is blank, the Name is used as the display name.

Description
An editable text field to provide explanatory text about the project.

Versions to keep
Automatically deletes the oldest committed project version when the number of versions exceeds the specified number. The default is 25.

Note: The cleanup process is not immediate, but runs in the background every 20 minutes.

Is public

A check box that indicates whether anyone can see the project (checked) or if users and groups must be explicitly added as members (cleared).

Click **Save** to keep the current state of the settings.

Sharing

You can share a project by adding users and groups as authors or viewers.

- Typing in the text box filters on users and groups with the search string in their name. Select the level of sharing and click **Add member** to add to the list of members.
  - Authors are full members of a project, and can modify the project as well as the folders and files within it. These users and members of these groups have write (Analytic Server Export node) access to this project when connecting to Analytic Server through IBM® SPSS® Modeler.
  - Viewers can see the folders and files within a project, and define data sources over the objects within a project, but cannot modify the project.
- To remove an author, select a user or group in the Author list and click **Remove member**.

Note: Administrators have read and write access to every project, regardless of whether they are specifically listed as a member.

Note: Changes made to Sharing are immediately and automatically applied.

Files

Project structure pane

The right pane shows the project/folder structure for the currently selected project. You can browse the folder structure, but it is not editable, except through the buttons.

- Click **Download file to the local filesystem** to download a selected file to the local file system.
- Click **Delete the selected file(s)** to remove the selected file/folder.

File Viewer

Shows the folder structure for the current project. The folder structure is only editable within defined projects. That is, you cannot add files, create folders, or delete items at the root level of the **Projects** mode. To create delete a project, return to the Project listing.

- Click **Upload file to HDFS** to upload a file to the current project/subfolder.
- Click **Create a new folder** to create a new folder under the current folder, with the name you specify in the **New folder name** dialog.
- Click **Download file to the local filesystem** to download the selected files to the local file system.
- Click **Delete the selected file(s)** to remove the selected files/folders.

Versions

Projects are versioned based on changes to the file and folder contents. Changes to a project's attributes, such as the description, whether it is public, and with whom it is shared, do not require a new version. Adding, modifying, or deleting files or folders does require a new version.

Project versioning table

The table displays the existing project versions, their creation and commit dates, the users responsible for each version, and the parent version. The parent version is the version upon which the selected version is based.

- Click **Lock** to make changes to the selected project version contents.

- Click **Commit** to save all changes that are made to a project and make this version the current visible state of the project.
- Click **Discard** to discard all changes that are made to a locked project and return the visible state of the project to the most recently committed version.
- Click **Delete** to remove the selected version.

User management

Administrators can manage the roles of users and groups through the Users page.

The content area is divided into **Details** and **Principals** collapsible sections.

Details

Name A non-editable text field that displays the name of the tenant.

Description
An editable text field that allows you to provide explanatory text about the tenant.

URL This is the URL to give to users to log in to the tenant through the Analytic Server console.

Status **Active** tenants are currently in use. Making a tenant **Inactive** prevents users from logging in to that tenant, but does not delete any of the underlying information.

Principals

Principals are users and groups that are drawn from the security provider that is set up during configuration. You can change the role of principals to be Administrators or Users.

Metrics

Allows you to configure resource limits for a tenant. Reports the disk space currently used by the tenant.

- You can set a maximum disk space quota for the tenant; when this limit it reached, no more data can be written to disk on this tenant until enough disk space is cleared to bring the tenant disk space usage below the quota.
- You can set a disk space warning level for the tenant; when the quota is exceeded, no analytic jobs can be submitted by principals on this tenant until enough disk space is cleared to bring the tenant disk space usage below the quota.
- You can set a maximum number of parallel jobs that can be run at a single time on this tenant; when the quota is exceeded, no analytic jobs can be submitted by principals on this tenant until a currently running job completes.
- You can set the maximum number of fields a data source can have. The limit is checked whenever a data source is created or updated.
- You can set the maximum number of records a data source can have. The limit is checked whenever a data source is created or updated; for example, when you add a new file or change settings for a file.
- You can set the maximum file size in megabytes. The limit is checked when a file is uploaded.

Security provider configuration

Allows you to specify the user authentication provider. **Default** uses the default tenant's provider, which was set up during installation and configuration. **LDAP** allows you to authenticate users with an external LDAP server such as Active Directory or OpenLDAP. Specify the settings for the provider and optionally specify filter settings to control the users and groups available in the Principals section.

Naming rules

For anything that can be given a unique name in Analytic Server, such as data sources and projects, the following rules are applied to those names.

- Within a single tenant, names must be unique within objects of the same type. For example, two data sources cannot both be named insuranceClaims, but a data source and a project could each be named insuranceClaims.
- Names are case-sensitive. For example, insuranceClaims and InsuranceClaims are considered unique names.
- Names ignore leading and trailing white space.
- The following characters are invalid in names.

  `~, #, %, &, *, {, }, \\, :, <, >, ?, /, |, ", \t, \r, \n`

Chapter 3. SPSS Modeler Integration

SPSS Modeler is a data mining workbench that has a visual approach to analysis. Each distinct action in a job, from accessing a data source to merging records to writing out a new file or building a model, is represented by a node on the canvas. We link these actions together to form an analytical stream.

In order to construct a SPSS Modeler stream that can be run against an Analytic Server data source, begin with an Analytic Server Source node. SPSS Modeler will push as much of the stream back to Analytic Server as possible, then, if necessary, pull a subset of the records to finish executing the stream "locally" in SPSS Modeler server. You can set the maximum number of records SPSS Modeler will download in the Analytic Server stream properties.

If your analysis ends with records written back to HDFS, finish the stream with an Analytic Server Export node.

See the SPSS Modeler documentation for details on these nodes.

Supported nodes

Many SPSS Modeler nodes are supported for execution on HDFS, but there may be some differences in the execution of certain nodes, and some are not currently supported. This topic details the current level of support.

General

- Some characters that are normally acceptable within a quoted Modeler field name will not be accepted by Analytic Server.
- For a Modeler stream to be run in Analytic Server, it must begin with one or more Analytic Server Source nodes and end in a single modeling node or Analytic Server Export node.
- It is recommended that you set the storage of continuous targets as real rather than integer. Scoring models always write real values to the output data files for continuous targets, while the output data model for the scores follows the storage of the target. Thus, if a continuous target has integer storage, there will be a mismatch in the written values and the data model for the scores, and this mismatch will cause errors when you attempt to read the scored data.

Source

- A stream that begins with anything other than an Analytic Server source node will be run locally.

Record operations

All Record operations are supported, with the exception of the Streaming TS and Space-Time-Boxes nodes. Further notes on supported node functionality follow.

Select

- Supports the same set of functions supported by the Derive node.

Sample

- Block-level sampling is not supported.
- Complex Sampling methods are not supported.

Aggregate

- Contiguous keys are not supported. If you are reusing an existing stream that is set up to sort the data and then use this setting in the Aggregate node, change the stream to remove the Sort node.

- Order statistics (Median, 1st Quartile, 3rd Quartile) are computed approximately, and supported through the Optimization tab.

Sort

- The Optimization tab is not supported.

In a distributed environment, there are a limited number of operations that preserve the record order established by the Sort node.

- A Sort followed by an Export node produces a sorted data source.
- A Sort followed by a Sample node with **First** record sampling returns the first *N* records.

In general, you should place a Sort node as close as possible to the operations that need the sorted records.

Merge

- Merge by Order is not supported.
- The Optimization tab is not supported.
- Analytic Server does not join on empty string keys; that is, if one of the keys you are merging by contains empty strings, then any records that contain the empty string will be dropped from the merged output.
- Merge operations are relatively slow. If you have available space in HDFS, it can be much faster to merge your data sources once and use the merged source in following streams than to merge the data sources in each stream.

R Transform

The R syntax in the node should consist of record-at-a-time operations.

Field operations

All Field operations are supported, with the exception of the Transpose, Time Intervals, and History nodes. Further notes on supported node functionality follow.

Auto Data Prep

- Training the node is not supported. Applying the transformations in a trained Auto Data Prep node to new data is supported.

Type

- The Check column is not supported.
- The Format tab is not supported.

Derive

- All Derive functions are supported, with the exception of sequence functions.
- Split fields cannot be derived in the same stream that uses them as splits; you will need to create two streams; one that derives the split field and one that uses the field as splits.
- A flag field cannot be used by itself in a comparison; that is, if (flagField) then ... endif will cause an error; the workaround is to use if (flagField=trueValue) then ... endif
- It is recommended when using the ** operator to specify the exponent as a real number, such as x**2.0, instead of x**2, in order to match results in Modeler

Filler

- Supports the same set of functions supported by the Derive node.

Binning

The following functionality is not supported.

- Optimal binning
- Ranks

- Tiles -> Tiling: Sum of values
- Tiles -> Ties: Keep in current and Assign randomly
- Tiles ->Custom N: Values over 100, and any N value where 100 % N is not equal to zero.

RFM Analysis

- The Keep in current option for handling ties is not supported. RFM recency, frequency, and monetary scores will not always match those computed by Modeler from the same data. The score ranges will be the same but score assignments (bin numbers) may differ by one.

Graphs

All Graph nodes are supported.

Modeling

The following Modeling nodes are supported: Linear, Neural Net, C&RT, Chaid, Quest, TCM, TwoStep-AS, STP, and Association Rules. Further notes on those nodes' functionality follow.

Linear When building models on big data, you will typically want to change the objective to Very large datasets, or specify splits.

- Continued training of existing PSM models is not supported.
- The Standard model building objective is only recommended if split fields are defined so that the number of records in each split is not too large, where the definition of "too large" is dependent upon the power of individual nodes in your Hadoop cluster. By contrast, you also need to be careful to ensure that splits are not defined so finely that there are too few records to build a model.
- The Boosting objective is not supported.
- The Bagging objective is not supported.
- The Very large datasets objective is not recommended when there are few records; it will often either not build a model or will build a degraded model.
- Automatic Data Preparation is not supported. This can cause problems when trying to build a model on data with many missing values; normally these would be imputed as part of automatic data preparation. A workaround would be to use a tree model or a neural network with the Advanced setting to impute missing values selected.
- The accuracy statistic is not computed for split models.

Neural Net

When building models on big data, you will typically want to change the objective to Very large datasets, or specify splits.

- Continued training of existing standard or PSM models is not supported.
- The Standard model building objective is only recommended if split fields are defined so that the number of records in each split is not too large, where the definition of "too large" is dependent upon the power of individual nodes in your Hadoop cluster. By contrast, you also need to be careful to ensure that splits are not defined so finely that there are too few records to build a model.
- The Boosting objective is not supported.
- The Bagging objective is not supported.
- The Very large datasets objective is not recommended when there are few records; it will often either not build a model or will build a degraded model.
- When there are many missing values in the data, use the Advanced setting to impute missing values.
- The accuracy statistic is not computed for split models.

C&R Tree, CHAID, and Quest

When building models on big data, you will typically want to change the objective to Very large datasets, or specify splits.

- Continued training of existing PSM models is not supported.
- The Standard model building objective is only recommended if split fields are defined so that the number of records in each split is not too large, where the definition of "too large" is dependent upon the power of individual nodes in your Hadoop cluster. By contrast, you also need to be careful to ensure that splits are not defined so finely that there are too few records to build a model.
- The Boosting objective is not supported.
- The Bagging objective is not supported.
- The Very large datasets objective is not recommended when there are few records; it will often either not build a model or will build a degraded model.
- Interactive sessions is not supported.
- The accuracy statistic is not computed for split models.

Model scoring

All models supported for modeling are also supported for scoring. In addition, locally-built model nuggets for the following nodes are supported for scoring: C&RT, Quest, CHAID, Linear, and Neural Net (regardless of whether the model is standard, boosted bagged, or for very large datasets), Regression, C5.0, Logistic, Genlin, GLMM, Cox, SVM, Bayes Net, TwoStep, KNN, Decision List, Discriminant, Self Learning, Anomaly Detection, Apriori, Carma, K-Means, Kohonen, R, and Text Mining.

- No raw or adjusted propensities will be scored. As a workaround you can get the same effect by manually computing the raw propensity using a Derive node with the following expression: if 'predicted-value' == 'value-of-interest' then 'prob-of-that-value' else 1-'prob-of-that-value' endif
- When scoring a model, Analytic Server does not check to see if all fields used in the model are present in the dataset, so be sure that's true before running in Analytic Server

R The R syntax in the nugget should consist of record-at-a-time operations.

Output

The Matrix, Analysis, Data Audit, Transform, Statistics, and Means nodes are supported.

The Table node is supported by writing a temporary Analytic Server data source containing the results of upstream operations. The Table node then pages through the contents of that data source.

Export A stream can begin with an Analytic Server source node and end with an export node other than the Analytic Server export node, but data will move from HDFS to SPSS Modeler Server, and finally to the export location.

Chapter 4. Troubleshooting

This section describes some common usage issues and how you can fix them.

Data sources

Filters defined on partitioned columns in HCatalog data sources are not honored

This is an issue seen in some versions of Hive, and can be seen in the following situations.

- If you define an HCatalog data source and specify a filter in the data source definition.
- If you create a Modeler stream with a Filter node that references the partitioned table column.

The workaround is to add a Derive node to the Modeler stream that creates a new field with values equal to the partitioned column. The Filter node should reference this new field.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Printed in USA