iWay Software

An Information Builders Company

iWay

iWay Big Data Integrator New Features Bulletin and Release Notes

Version 1.5.0

DN3502232.1216

Contents

iWay Big Data Integrator Version 1.5.0

This document describes new features and provides release notes for iWay Big Data Integrator (iBDI) version 1.5.0. It is intended for all levels of users, including system integrators, application developers, and administrators.

Topics:

New Features

In this section:

This section provides a summary of the new features for iWay Big Data Integrator (iBDI) version 1.5.0.

- **Spark Pipeline**

  You can configure a Spark-based pipeline in iBDI to transform, cleanse, join, or perform other operations on incoming data. The pipeline in iBDI uses the DataFrames API of the Apache Spark Framework, which is an optimized engine for cluster computing. A DataFrame is a data set that is organized into named columns.

- **Kafka**

  Kafka is a clustered system that streams records in *topics*. Producers publish stream records and consumers subscribe to the stream topics, and can collect data or perform further operations on the data.

- **Nifi**

  Nifi supports an abstraction called a *FlowFile* that can be used to wrap content, and a *Processor* to work on the data. Nifi can be used to extend the types of data and sources that can be ingested, and to display their flows.

- **Streaming**

  Spark streaming is supported with Avro schema, XML and JSON objects, native Spark streaming, Kafka streaming, and Nifi streaming.

- **RStat Model Integration**

RStat models can be incorporated into pipeline compute objects.

- **In Memory Computing**

  Using the Spark-based pipeline, many operations can be performed *in memory* on the Spark cluster, which is more efficient and saves resources, time and read/write operations.

- **Lambda Architecture**

  Using Kafka and Spark-based pipelines, the Lambda Architecture provides a data warehouse solution with batch operations for bulk data retrieval and real time queries on a speed layer so that important up-to-date data is not missed.

- **Flafka (Flume and Kafka)**

  Flume reads from a source and writes to a Kafka topic. This allows for load balancing as Kafka is multi-streamed for data distribution to multiple destinations, and Flume reads streams.

Spark Pipeline

A pipeline can be thought of as a chain of stages that are connected from a *source* to a *target*. Data is transferred between stages in the pipeline. Each stage may perform a specific operation on the data (for example, *Show* or *Transform*).

A pipeline may have multiple *compute* stages between a source and a target. Each compute stage and the target have an input and an output schema associated with them. The source has no input schema, as the input schema is available from the data source itself. Because each node has an associated schema, intelligent operations can be performed on the data for the stage, such as validation, building expressions, and Where clauses.

Pipelines can be used for any number of operations, including data transformation, in memory computing, data cleansing with iWay Data Quality Server (DQS), modeling with RStat, and other operations.

A pipeline has a single execution path, without branching or looping. Pipelines can be chained together, so that the target stage for one pipeline becomes the source stage for the next pipeline for multiple or complex operations on data.

The Spark-based pipeline is the preferred processing mode in iBDI version 1.5.0. While iBDI version 1.4.0 and earlier versions were based on the Hadoop MapReduce framework, and these operations are still supported in the product, it is recommended to use the pipeline interface moving forward for its resiliency, ease of operation, and improved processing speed.

Kafka

Kafka is a publish and subscribe clustered messaging system. Unlike traditional messaging systems, Kafka is fully supported by the distributed aspect of Hadoop, partitioning the topic between stages and performing parallel operations on the stages.

Nifi

Nifi is used with iBDI to increase the number of operations that can be performed and the types of data that can be ingested. Nifi introduces the abstractions of FlowFile for data input, and Processor for operations on the input. Nifi can also be used to display pipelines.

Streaming

In this section:

Streaming can be accomplished through pipelines or native Spark, Flume support, and other techniques. XML or JSON flat objects or data with Avro schema are supported.

Spark Streaming

Since the structure of streaming data is unknown, the user can provide an Avro schema to define it. Even if the data is not in Avro format, the metadata can be used to build a schema for the DataFrame. If the data format is Avro, iBDI will use the schema at runtime to read the streaming data.

XML and JSON Data Streams

If the incoming stream data is XML, iBDI will use the fields in the Avro schema to search for elements in the XML document. If the element is found, then its value will used as the value of that column in the data frame. JSON data works similarly in that the fields in the Avro schema are used to search for fields in the JSON object. iBDI only supports flat JSON/XML objects. JSON/XML objects with complex, multi-level structures are not supported.

Native Spark Streaming

A simple socket text stream that reads data from a port and builds data frames to pass to the pipeline.

Kafka Streaming

Reads data from a Kafka topic. You must specify a consumer group. Kafka will send the last message to a pipeline within a consumer group. Utilizing consumer groups will enable load balancing between pipelines.

Nifi Streaming

Configure a Nifi pipeline listener with the same name given to the Nifi outbound port. This will enable the pipeline to stream data from Nifi.

RStat Model Integration

Using the WebFOCUS RStat 2.0 modeling tool, a model can be exported as a JAR file. This JAR file can be built through Maven, provisioned to iBDI, and then deployed into a pipeline.

Transformer

Hive tables in a pipeline have a transform associated with them. A transform allows a source table to be mapped to an output schema. The source table can be joined to other tables before the output schema.

The output schema may have functional expressions added, WHERE clause additions, and different column mapping than the source or Join tables.

The output schema target may be persisted or used for later stage processing in a pipeline.

Flume

Flume now supports Kafka Sink as an output type. Kafka support allows for messages to be passed and type transformation.

Flafka

Flume using Kafka Sinks is referred to as *Flafka*, which uses the buffered capability of Flume with the messaging ability of Kafka to provide high quality online messaging operations.

- Active MQ support.
- Available through the JMS option.

Run Configurations Facility

Support for pipeline types as Deploy (run online) and Publish (run as a job) options.

Multiple Sqoop and Flume jobs can be used in a Publish operation.

Lambda Architecture

This is a technique where Spark configurations can be used with a batch and online update channels for fast and reliable query capabilities. The Batch layer manages the master data set with an append data update based on batch views. The Speed layer deals with fast queries based on recent data. iBDI can be used to implement a Lambda Architecture query implementation with the new Spark, Kafka, and Flume components.

Release Notes

In this section:

This section provides release notes for iWay Big Data Integrator (iBDI) version 1.5.0.

Installation

If you are uninstalling a previous version of iBDI before installing the latest version (1.5.0), it is recommended to reboot the Windows operating system. Some file handles may not be released even if iBDI is uninstalled. If iBDI does not start after you install version 1.5.0, reboot Windows and restart iBDI.

Data Sources

In this section:

There is an issue with *Import Connections* not importing the driver template. As a workaround, you must recreate the data source configuration.

ODA Data Sources

ODA data sources are not supported for iBDI version 1.5.0 input sources. Valid input sources are RDBMS, Streaming, and defined Hadoop sources (HDFS and Hive).

Data Source Explorer

- There is a known issue where the Data Source Explorer sorts table columns in inverse alphabetical order. As a workaround, determine the correct column order by navigating to the schema and table, then right-click and select *Sample Contents*. In the console window, a frame will be opened with the first 50 records in the column order returned by the current driver.
- Care should taken with the *Extract* command of the table viewer. This option extracts all records of the source table to a local file. In the case of a Hive table, the number of records could be very large. Use the *Sample Contents* option to view the first 50 records of the table in the viewer.

Pipelines

- In iBDI version 1.5.0, the Pipeline Builder is implemented by working with a single pipeline at a time. The Run Configurations facility supports only a single pipeline. Multiple pipelines can be created and edited in the builder workbench.
- Missing help text for pipeline parameters. For more information, see *Appendix A, Pipeline Parameters Reference* in the *iWay Big Data Integrator User's Guide*.
- When a pipeline is copied, an *invalid target specification* error may be listed in the Problems tab. If this occurs, change a parameter value in the pipeline to activate the Save button. Then save the pipeline again.
- The Problems tab in the lower pane reflects the current state of the constructed pipeline. Any problems that are listed here remain until the pipeline is saved by clicking *Save*, or until the problem is corrected. A pipeline that has a problem cannot be sent to the Run Configurations facility for execution or to be published.

- A pipeline deployment fails on the initial run when pointing to a new user's HDFS directory. For example:

  `/user/<user>`

  As a workaround, manually create the directory structure.

Pipeline Transformer

- When you configure a Join using the Transformer, only unique column names can be mapped to the output schema. This is currently a limitation, which will be resolved in the next release.
- The initial transform source displays *Edit Transform* because every Hive source has a default transform. If the transform is reset, then *Add Transform* is displayed as there is no transforms available after a reset.
- The Output Schema target contains all of the mappings. Hover the mouse pointer over the Source table to enable the icons or use the draw palette. Draw a mapping line from the Source to the Target to connect the tables and begin the mapping operation. In the Mapper dialog, select the columns to map to the output target by clicking (or using the *Ctrl* key for multiple fields) and dragging them to the target column.

  When multiple tables are mapped to the Output Schema, click the line that represents the mapping from the Source to the Target to view the mapping lines for that table. Only mapping lines for one table at a time are displayed in the map target list. The other columns are present, but do not display their mapping lines unless their source table is the current table that is selected.

General Expression Builder and Where Clause Expression Builder

The general Expression Builder shows all of the possible Hive operators. However, the expression should return an aggregate value or transformation. The Where Clause Expression Builder is available only on the Output Schema target, and must return a Boolean value. Each operator has a brief description of the available operator.

Supported operators include:

- Relational operators (=, ?, ==, <>, <, >, >=, <=, etc.)
- Arithmetic operators (+, -, *, /, %, etc.)
- Logical operators (AND, &&, OR, ||, etc.)
- Complex type constructors
- Mathematical functions (sign, ln, cos, etc.)
- String functions (instr, length, printf, etc.)

Transform Mapper and Join Tool

An issue can occur with multiple revisions of a Join mapping and a Where clause, which generates an *invalid column 0* error.

As a workaround, reset the Where clause by clicking the Erase icon. Save the transform and then specify the Where clause again if this error occurs.

Input Document Specification

Currently, only version 1 of the Input Document Specification (IDS) is supported.

Sqoops

When you run a Sqoop configuration in the Run Configurations facility, if you set the value in the Deployment Location field to a new multi-level directory path (for example, `dir1/dir2/dir3/deployment_name`), a *No such file or directory* error is generated.

As a workaround, you must manually create the multi-level directory structure. This issue will be resolved in the next release of iBDI.

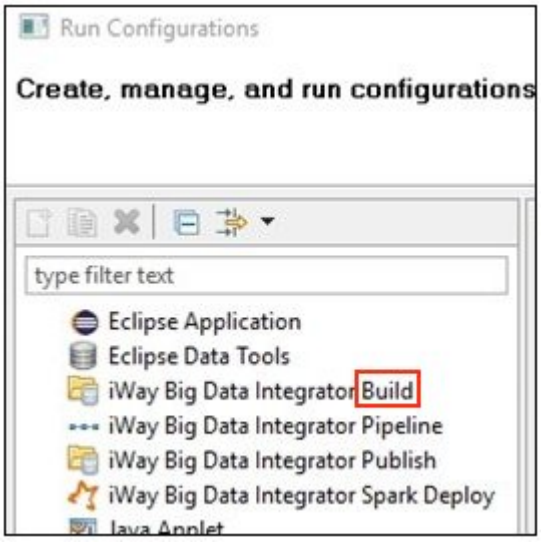
Wranglers

A wrangler requires a file on HDFS. Before invoking a wrangler, create a link to an HDFS by either clicking the *Hadoop* icon on the toolbar or click *File*, select *New*, *Hadoop*, and then *New HDFS Server* from the main menu.

After the connection is specified and connected, invoke the wrangler to create a new table.

Run Configurations Facility

There is a terminology (labeling) error in the Run Configurations facility, as shown in the following image.



The *iWay Big Data Integrator Build* option should be labeled as *iWay Big Data Integrator Deploy*.

If you need to **deploy** an iBDI project component (for example, Pipeline, Sqoop, Mapper, etc.), then select the *iWay Big Data Integrator Build* option.

Reader Comments

In an ongoing effort to produce effective documentation, the Technical Content Management staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please share your suggestions for improving this publication and alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail:	Technical Content Management Information Builders, Inc. Two Penn Plaza New York, NY 10121-2898
Fax:	(212) 967-0460
Email:	*books_info@ibi.com*
Website:	*http://www.documentation.informationbuilders.com/connections.asp*

Name:_____

Company:_____

Address:_____

Telephone:_____ Date:_____

Email:_____

Comments:

Reader Comments