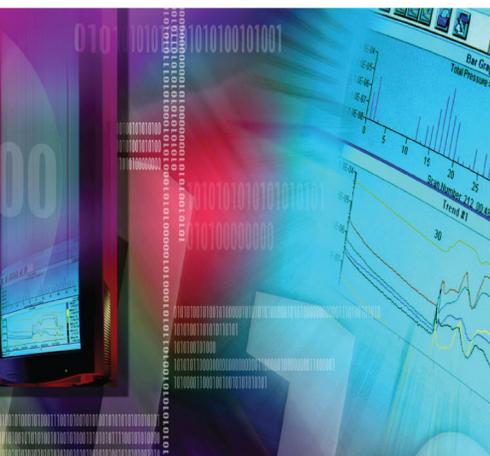


alteryx



# Alteryx and Revolution Analytics Integration Guide

v 1.3, November 2015

## Overview

This document covers the integration of Revolution Analytics' R-based Revo ScaleR hybrid disk/memory approach for scaling predictive analytics into Alteryx. The integration is implemented through the creation of a Revolution Analytics XDF format file, which triggers a number of Alteryx predictive modeling tools (Linear Regression, Logistic Regression, Count Regression, Gamma Regression, Stepwise, Decision Tree, Forest Model, Lift Chart, and Score) to make use of Revolution Analytics scalable Revo ScaleR algorithms.

In what follows, the Alteryx / XDF environment will be described in greater detail.

## The Alteryx / XDF Environment

There are two tools provided to perform read and write functions for XDF files in Alteryx, XDF Output and XDF Input. The XDF Output tool takes an Alteryx data stream and writes it to an XDF file either in Alteryx's temporary directory or to a user specified permanent location on disk. In addition to writing the XDF file, an "XDF metadata stream" is also produced. The XDF metadata stream provides downstream predictive tools with information about the underlying metadata describing the data, along with information that enables a predictive tool to determine the location of the relevant XDF file. Figure 1 provides a simple sample workflow incorporating the use of the XDF Output tool, while Figures 2 and 3 give an example XDF metadata stream in terms of both the data in the stream (Figure 2) and the the metadata contained in the stream (Figure 3).

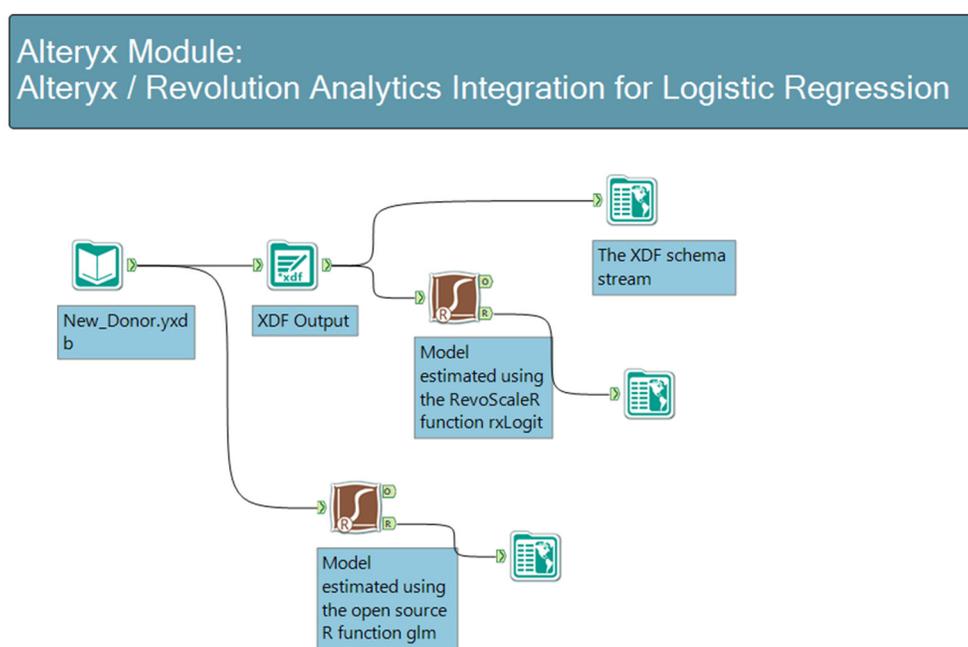


Figure 1. An Alteryx Workflow with Revo ScaleR Integration

Record #	Donate	Degrees	First_Years	Last_Years	Undergraduate	First_School	Last_School	Faculty_Staff	Intercollegiate	Intramural
1	No	2	21	16	Yes	SC	NU	No	No	No
2	No	2	27	25	No	AR	AR	No	No	No

Figure 2. The Data of an XDF Metadata Stream Data

	Name	Type	Size	Source
1	Donate	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
2	Degrees	Int32	4	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
3	First_Years	Int32	4	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
4	Last_Years	Int32	4	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
5	Undergraduate	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
6	First_School	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
7	Last_School	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
8	Faculty_Staff	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
9	Intercollegiate	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
10	Intramural	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
11	Other_Activities	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
12	Gender	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
13	Child	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
14	Parent	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
15	Spouse	V_WString	1073741823	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
16	Telephone	Int32	4	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
17	Mail	Int32	4	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
18	Personal	Int32	4	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}
19	Combined	Int32	4	{"Context":"XDF","File.Loc":"C:/Users/dputler/Work/Example_Data/New_Donor.xdf"}

Figure 3. The Data of an XDF Metadata Stream Metadata

As is indicated in Figure 1, the input into a predictive modeling tool from an XDF Output tool (shown in the upper portion of the canvas) is an XDF metadata stream, which causes a predictive modeling tool (in this case the Logistic Regression tool) to identify the input as being an XDF file, resulting in the model being estimated using the appropriate Revo ScaleR modeling function (in this case Revo ScaleR's rxLogit function). Instead, if the input into an Alteryx predictive modeling tool is a normal Alteryx data stream (as opposed to an XDF metadata stream), then the appropriate open source R function is used (in this case R's glm function). The information the user is asked to provide in the Alteryx predictive modeling tool is the same in both cases, with the use of the Revo ScaleR or open source R modeling function being determined by the type of input the tool receives.

As is illustrated in Figure 2, an XDF metadata stream consists of the number of fields the user has selected for inclusion into the XDF file in the XDF Output tool and two data records. This small amount of data is enough to properly populate the user interface of downstream predictive modeling tools. Additional metadata (as shown in Figure 3) is conveyed via a JSON string that contains information about the compute context (with a keyword of “Context” and a value of “XDF” to work with a standalone XDF file) and the path to the XDF file to be used in the analysis (with a keyword of “File.Loc” and a string of the full file path to the XDF file as the value) which is contained in the “Source” field of the metadata. The JSON string is repeated for each field to insure that it is available to a downstream modeling tool regardless of what fields are included in the model. The metadata is read by a modeling tool, allowing for model estimation. The use of the JSON string to convey information beyond the schema of the data table will also be used for future in-database / in-Hadoop applications, but altered to contain other information (e.g., the type of database, the controller’s address, the data to use, authentication information, etc.).

Saving the XDF file to disk is useful if the user plans on creating several Alteryx workflows using the same XDF file. In subsequent workflows that use the XDF file, the XDF Input tool is used to read needed information into Alteryx. If the XDF data will only be used with predictive modeling tools (so no further data cleaning, blending, or other augmentation is to be undertaken by the user), then only the XDF metadata stream for the XDF file needs to be read into Alteryx (saving considerable time). Otherwise, the full XDF file will be read into Alteryx. Whether only the metadata stream or the full XDF file is read into Alteryx is specified by the user via a radio button option in the XDF Input tool, as is shown in Figure 4. The tool makes use of Alteryx’s new capability of reading data into R in chunks, and the user is asked to provide the number of rows to read in each chunk. The tool defaults to 256,000 records, which was selected to make use of the number of rows in an Alteryx block (64,000 rows) and the optimal number of rows to use with Revo ScaleR functions (between 200,000 and 300,000).

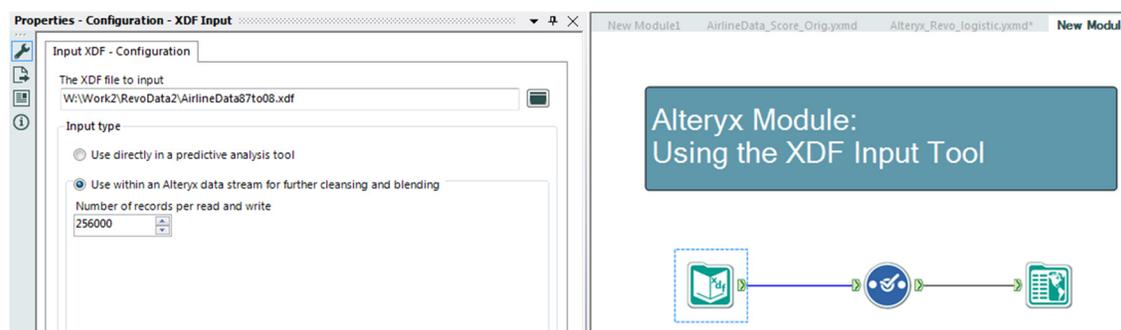


Figure 4. Use of the XDF Input Tool

Scoring in the Alteryx/XDF environment uses the same mechanism as scoring in the traditional open source R Alteryx environment, with the exception that if the data to be scored is in an XDF file, the scored values will be appended to this file. Otherwise, the Revo ScaleR based models will append predicted values to the Alteryx data stream used for scoring. What enables scoring to scale is Alteryx's ability to read and write data between R and Alteryx a chunk at a time, regardless of whether or not the model used in scoring was created using Revo ScaleR or open source R tools. We have run tests where over 160MM records were scored in Alteryx using only open source R tools.

## Integrating Revolution R with Alteryx Designer

Follow the steps below to integrate Revolution R with Alteryx Designer.

1. Install the current RRE version of Revolution R.  
*Important: Revolution R must be installed before the Alteryx Designer Predictive Tools are installed.*
2. In Alteryx Designer click *Options > Download Predictive Tools* (or go directly to <http://downloads.alteryx.com/predictive.html>).
3. Select the **Alteryx Predictive Tools for Revolution Analytics** installer for the version of RRE that has been installed on the machine.
4. Verify that Revolution RRE was installed correctly by verifying the **XDF Input** and **XDF Output** tools are available in the In/Out tool category.

