Revolution R Enterprise 6.0.1 for README

Revolution R Enterprise 6.0.1 for 32-bit and 64-bit Windows and 64-bit Red Hat Enterprise Linux (RHEL 5.x and RHEL 6.x) features a patch release of the RevoScaleR package that provides fast, scalable data management and data analysis: the same code scales from data frames to local, high-performance .xdf files to data distributed across a Windows HPC Server cluster or IBM Platform Computing LSF cluster. RevoScaleR also allows distribution of the execution of essentially any R function across cores and nodes, delivering the results back to the user.

Installation instructions and instructions for getting started are provided in your confirmation e-mail.

What's New in Revolution R Enterprise 6.0.1

Changes to Threading

- New RevoScaleR functions `rxSetEnableThreadPool` and `rxGetEnableThreadPool` can be used to set and get the current thread pool state on Linux. By default, the thread pool is disabled on Linux to allow R sessions involving RevoScaleR computations to be forked successfully. Users who are sure they will not be using fork to spawn new processes involving RevoScaleR can enable the thread pool by calling `rxSetEnableThreadPool=TRUE` after loading RevoScaleR.
- Earlier versions of Revolution R Enterprise used a mixture of threading models. Revolution R 6.0.1 uses only the gcc threading model.

Changes to Distributed Computing

- The `continueOnFailure` argument to `rxExec` now defaults to `TRUE`.
- The Azure Burst compute context now uses https instead of http to communicate between the desktop and Azure nodes.
- The way in which remote R processes are terminated was changed in HPC Server and Azure Burst compute contexts to avoid occasional failures to return state correctly.
- The `rxGetInfo` and `rxGetVarInfo` functions now work correctly in distributed compute contexts when called within a function.
- The `rxCancelJobs` function now works correctly in LSF compute contexts.
- `RxLocalParallel` on Linux now uses a snow-like backend, rather than a multicore-like backend that could hang R when used with an HPA function such as `rxGlm` or `rxLinMod`.
- `RxLsfCluster` compute contexts now use `$TEMP` as the default working directory if `$HOME` does not exist or is not writable.
- The `rxGetJobs` function now works correctly if `exactMatch=TRUE`.

Changes to Data Analysis

- The `rxPredict` function now works with non-xdf data sources as long as predictions are written to an .xdf or returned as a data frame, and now works with a data frame when only one variable is used in the model.

Copyright © 2012 Revolution Analytics. All rights reserved. Revolution R, Revolution R Enterprise, RPE, RevoScaleR, RevoDeployR, NetWorkSpaces, NWS, ParallelR, and Revolution Analytics are trademarks of Revolution Analytics. All other trademarks are the property of their respective owners

- The `rxQuantile` function now works with non-xdf data sources, and now has `fweights` and `pweights` arguments.

Changes to Data Import and Manipulation

- For fast text import, `rowsToSkip` with the first row as column names no longer imports all data as character data.
- Transformed variables are now returned if the input data is a data frame and `writeModelVars=TRUE`.

Known Issues:

- ***Known Issues in Revolution R Enterprise 6.0.1***

Copyright © 2012 Revolution Analytics. All rights reserved. Revolution R, Revolution R Enterprise, RPE, RevoScaleR, RevoDeployR, NetWorkSpaces, NWS, ParallelR, and Revolution Analytics are trademarks of Revolution Analytics. All other trademarks are the property of their respective owners