Oracle Retail® Extract Transform and Load™ 11.3

Release Notes

Corporate Headquarters:

Oracle
950 Nicollet Mall
Minneapolis, MN 55403
USA

888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000

Fax:
+1 612 587 5100

European Headquarters:

Oracle
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

Sales Enquiries:
+44 (0)20 7563 46 46

Fax:
+44 (0)20 7563 46 10

Information in this documentation is subject to change without notice.

Oracle Retail provides product documentation in a read-only-format to ensure content integrity. Oracle Customer Support cannot support documentation that has been changed without Oracle authorization.

The functionality described herein applies to this version, as reflected on the title page of this document, and to no other versions of software, including without limitation subsequent releases of the same software component. The functionality described herein will change from time to time with the release of new versions of software and Oracle reserves the right to make such modifications at its absolute discretion.

Printed in the United States of America.

Customer Support

Customer Support hours

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Oracle customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method	Contact Information
E-mail	support@retek.com
Internet (ROCS)	rocs.retek.com Oracle Retail's secure client Web site to update and view issues
Phone	+1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835

Mail

Oracle Customer Support
950 Nicollet Mall
Minneapolis, MN 55403

When contacting Customer Support, please provide:

- Product version and program/module name.
- Functional and technical description of the problem (include business impact).
- Detailed step-by-step instructions to recreate.
- Exact error message received.
- Screen shots of each step you take.

Overview

This RETL 11.3 release is a feature release containing the following enhancements:

- New database operators INSERT, UPDATE, DELETE, and PREPAREDSTATEMENT
- New EXIT operator
- Java API interface
- Support for parentheses and complex expressions in FILTER
- Improved OUTPUT syntax in PARSER

See the RETL 11.3 Programmer’s Guide for more information.

Bug fixes

Many bugs were fixed in RETL 11.3. See the ‘Changes since RETL 11.2.2’ section below.

Note: Several bug fixes catch errors in flows that were not detected in earlier versions of RETL. Before upgrading to RETL 11.3, Oracle Retail recommends that you perform a full regression test. Pay special attention to the following bugs:

- **Bug #1480:** SORTFUNNEL and FUNNEL can set a string field’s “maxlength” improperly.
- **Bug #1319:** Error in NULL field comparisons in FILTER.

Database support

Database support for DB2 and Teradata has been dropped for this and all prior versions of RETL.

See the ‘Compatibility matrix’ section below for a list of all supported platforms and databases.

Product certification

RETL 11.3 has been certified with RDW 10.2.5. See the ‘Certification’ section below for a matrix of Oracle Retail products and the certified version of RETL.

Oracle Retail strongly recommends that, before any installation or development work is performed, the RETL 11.3 Programmer's Guide is read in its entirety. In particular, read and understand “Chapter 2 – Installation and system configuration”. In addition, pay special attention to the “Known issues” section of these Release Notes.

Product groups should take care to complete all certifications and regression/volume testing with RETL 11 versions before deployment to customer sites.

The build number for RETL 11.3 is 722. After a successful installation of RETL 11.3, running RETL with the –v command line option should produce the following output:

```
$ retl –v

retl 11.3 build 722
```

Clients that took delivery of prior releases and/or subsequent patch releases are strongly encouraged to upgrade to the RETL 11.3 patch release once a full regression test has been performed. Issues identified in RETL 11 releases and prior releases are resolved with enhancements and fixes being delivered through RETL 11 patch releases. No additional releases of RETL 10.x are planned.

RETL 11.x enhancements

Performance enhancements

- Partitioning/sizing implementation

  The concept of partitioning is to parallelize as much as possible, which in RETL, ideally results in a flow uninhibited by bottlenecks. The goal is to code flows so that they can be partitioned to perform within a customer's batch window.

- New architecture

  RETL 11 releases include a new architecture that optimizes pipeline parallelism by consolidating connected operators into the same pipeline where possible. This architecture reduces the number of threads required, minimizing context switching and thrashing among threads. The framework thus supports improved performance.

- ORAREAD performance enhancements

  There have been significant improvements to performance in ORAREAD.

- Faster datasets

  The performance of the datasets that connect operators has been improved.

Error handling

- RETL 11 releases introduce better flow debugging by giving line and column specific error messages when RETL encounters problems reading an XML flow definition. Additionally, there is exception handling in RETL 11 that allows for better error handling and debugging.
- Due to the configuration capabilities of the error threshold in ORAWRITE, RETL can be configured to abort when reaching the maximum # of rejected records from SQL*Loader.
- There are overall improved error messages in 11.1 as opposed to the 10.x versions.

Debugging and logging

- RETL visual flow graphs

  Beginning with RETL 11.2, a command-line option can be used to generate a visual graph of a RETL flow, complete with operator names, link names, and thread identifications. See the RETL Programmer’s Guide for more information.

- Enhanced performance logging

  RETL 11.2 logs significant events such as query execution, sorting via gsort, loading data via native database load utilities (for example, SQL*Loader), and more.

- Performance hotspots/bottleneck graphs

  RETL now exports an HTML graph of performance activity. This graph uses visual cues to point out bottlenecks in the flow so that flow developers, performance analysts, and services consultants can diagnose the cause of slow-performing flows.

- Enhanced logging

  RETL 11 uses the log4j logging facility for all terminal and performance log output. The log4j facility is an extremely flexible open-source package maintained by the Apache Software Foundation. See http://logging.apache.org/log4j/docs/documentation.html and the RETL Programmer’s Guide for more information on log4j.

- Documentation

  Much-improved Programmer's Guide and Performance Tuning Guide.

- Online help

  A command-line option has been added to RETL 11 that allows flow developers to view operator syntax and usage via the command-line without having to refer to this document.

- Simplified installation and configuration

  RETL 11 releases consolidate code into a single binary rather than the 26 binaries of 10.x release. This design considerably eases installation. In addition, there is less environment setup that needs to be done in order to complete an installation.

- True platform independence

  The new architecture of the RETL 11 releases was written entirely in Java. As a result, RETL may now run on platforms that previously would have required significant efforts to port to. See the Programmer's Guide and the Compatibility Matrix for supported platforms. As a direct result of the platform independence, significant RETL resources in platform maintenance will be freed up to develop new features going forward.

- Backwards compatibility with previous releases

  A requirement of RETL 11 versions is that they be backward-compatible with the 10.x versions. However, RETL 11 versions are more strict on enforcement of valid XML flow interfaces, input schemas, and data fields. In some instances, the RETL 11 product will find data and/or flow errors that may have previously been unreported. The result may be more discarded or rejected records and/or error messages than previously identified.

New operators

- PARSER operator

  Business logic can be coded in a Java-like code script, which eliminates proliferation of too many operators, improves performance by reducing the number of operators (for example, FIELDMOD, BINOP, FILTER, SWITCH, COPY), reduces maintenance costs by decreasing complexity of flows, and aids in a better sizeability when partitioning flows.

- DBLOOKUP operator

  This operator allows the flow to join records directly to a table in the database.

- CHANGECAPTURELOOKUP operator

  Identical in functionality to CHANGECAPTURE, but can take unsorted input, which lends itself to better partitioning and sizing.

- SPLITTER operator

  The SPLITTER operator is a round-robin-based partitioning operator that allows equal distribution of records to each partition.

- INSERT operator

  The INSERT operator inserts records into an Oracle database with values from the current RETL record.

- UPDATE operator

  The UPDATE operator updates records in an Oracle database with values from the current RETL record.

- DELETE operator

  The DELETE operator deletes records from an Oracle database based on key fields in the current RETL record.

- EXIT operator

  The EXIT operator tracks the number of records passing through the operator and shuts down the flow when a specified limit has been reached.

Compatibility matrix

The following table shows the OS/database combinations that are supported and have been certified to work properly.

OS	Version	Arch	DB	Version
AIX	5.1	32/64	none	None
AIX	5.1	64	Oracle	9.2
AIX	5.2	32/64	none	None
AIX	5.2	64	Oracle	9.2
HP-UX	11.11	32/64	none	None
HP-UX	11.11	64	Oracle	9.2
HP-UX	11.11	64	Oracle	9.0.1
HP-UX	11.11	32	Oracle	8.1.7
HP-UX	11.23	IA64	None	None
HP-UX	11.23	IA64	Oracle	9.2
Solaris	8	32/64	none	None
Solaris	8	64	Oracle	9.2
Solaris	8	32/64	Oracle	9.0.1
Solaris	8	32	Oracle	8.1.7
Solaris	8	32	DB2	8.1.3
Solaris	9	32/64	none	None
Solaris	9	9	Oracle	9.2
Solaris	9	32/64	Oracle	9.0.1
Solaris	9	32	Oracle	8.1.7

Note: RETL 11 only requires one binary and installation for all database/platform combinations listed above.

Certification

The following table lists products that have been certified to work with a specific RETL version.

Product	RETL Version
RMS 11.0.2	RETL 11.2.1
RPM 11.0.1	RETL 11.2.1
ReIM 11.0.2	RETL 11.2.1
RDW 11.0	RETL 11.2.1
RDW 10.2.3	RETL 11.2.2
RDW 10.2.5	RETL 11.3

Installation

See the RETL Programmer's Guide for installation and configuration instructions.

Package contents

The following is a brief description of the contents of retl.11.3.tar.gz:

bin/

Includes executables required for running retl on different platforms.

File	Description
gsort.SunOS	gsort for SunOS
gsort.HP-UX	gsort for HP-UX
gsort.AIX	gsort for AIX
retl	RETL executable
rfx	Symbolic link to retl provided for backwards compatibility
verify_retl	The RETL verification tool
README.verify_retl	The README for verify_retl
install_graphing.ksh	Installs the GraphViz graphing package
install_jre.ksh	Installs the Java Runtime Environment
test_funcs.ksh	Used by verify_retl to run test flows

Note: There is only one binary (retl/rfx) required for all database and platform combinations

lib/

These are jar files that are needed in order to run RETL.

File	Description
retl.jar	RETL runtime library
log4j-1.2.9.jar	log4j Logging Facility
ojdbc14.jar	Oracle JDBC driver
xercesImpl.jar	Xerces XML Parser
xml-apis.jar	XML APIs
xmlParserAPIs.jar	XML APIs
tools.jar	Future enhancements
activation.jar	Future enhancements

etc/

This directory contains the documentation associated with RETL.

File	Description
rfx.conf	The default retl configuration file.
Logger.conf	The default logger configuration file.

samples/

This directory contains the samples packaged with RETL. See the README in the samples/ directory for more information about the samples packaged with RETL. These are the samples that are run after the install, by verify_retl.

graphing/

This directory contains the Graphviz application suite for graphical visualization of RETL flows. See the ‘Producing graphical output of flows with RETL’ section of the RETL Programmer's Guide.

JRE/

This directory contains the Java Runtime Environment (JRE) required by RETL.

Known issues

General known issues

- RETL does not handle arbitrary precision math.
- The export operator defaults to use '|'-delimited fields if a “schemafile” property is not specified. This issue can cause problems if the incoming data contains the ‘|’character. The recommendation is to use an export schemafile if it is possible for data to contain a ‘|’ character.
- JOIN operators may use a large amount of memory when the distribution of key-equivalent records is low and the number of records is high. It is important to test join operations with appropriate data volumes on each input side to verify the maximum amount of memory your application may require.

Differences between the 10.x and 11.x versions of RETL

- RETL 11 versions are stricter on the enforcement of valid XML flow interfaces, input schemas, and data fields. In some instances, the RETL 11 product finds data and/or flow errors that may have previously been unreported. The result may be more discarded or rejected records and/or error messages than previously identified.
- The output field order between 10.x and 11.x may be different for those flows that do not specify a “schemafile” property in the export operator.
- RETL 11.x versions require additional properties for the ORAREAD and ORAWRITE operators. “hostname” and “port” are now required properties and may be specified in rfx.conf for convenience. See the RETL Programmer's Guide for more information on syntax and usage.
- When upgrading an Oracle database, a separate instance of RETL should be used in order to test the new database. This recommendation is due to the fact that only one port can be specified in rfx.conf at a time. The other option is to use RETL with the '-c' option to use a different config file. See the RETL Programmer's Guide for more information.
- RETL 11.x versions may require more physical memory for certain flows than 10.x releases.

Backwards compatibility issues with 10.x versions

- Oracle Retail does not recommend upgrading to RETL 11 unless the product using RETL has been certified on RETL 11.

Known defects in RETL 11.3

- Bug #611 – Tests for table existence can return false matches if the user does not have sufficient privileges on a table.
- Bug #659 – A pipe delimiter is used when paging datasets to disk. This bug causes problem re-importing if there are pipes in the data.
- Bug #744 – SORT does not work with newlines in data.
- Bug #759 – Invalid INPUT element results in StackOverflowError in flow parsing.
- Bug#820 – EXPORT file schema can be broken when the input data is cached for numeric values and the export field length is less than the imported field length.
- Bug #941 – Parallel direct datafile ignored when partitioning.
- Bug #959 – Loops in INPUTs and OUTPUTs cause infinite loop in schema resolution.
- Bug #1101 – RFX_SHOW_SQL output cannot be shown without RFX_DEBUG output.
- Bug #1155 – “removedup” property of SORT does not work correctly when “numsort” > 1.
- The “statement” property of the PREPAREDSTATEMENT operator must use at least one field from the input schema.

Changes

Changes since RETL 11.2.2

Bug #781 – Duplicate OUTPUTs causes a warning then a NullPointerException

In prior versions of RETL 11, an operator with the same output dataset listed twice first issued a warning.

Starting in RETL 11.3, the following error is reported:

```
FATAL - E391: "dataset.v" registered more than once as INPUT for
generator:1.
```

Bug #795 – Too few INPUTs to a JOIN operator gives a bad error message

In prior versions of RETL 11, specifying too few inputs to any of the JOIN operators (INNERJOIN, LEFTOUTERJOIN, and so on) caused an ArrayIndexOutOfBoundsException.

Starting in RETL 11.3, an error is reported:

```
FATAL - E135: Exception in operator [innerjoin:1]
```

```
E369: JOIN operators require 2 inputs.
```

Bug #805 – GENERATOR sequence attributes not fully validated

In prior versions of RETL 11, invalid attributes in a sequence field defined in the “schema” property of GENERATOR were not detected.

Starting in RETL 11.3, a warning is displayed:

```
WARN  - W144: Sequence field in operator [generator:1] contains an
unrecognized attribute "attribute-name".
```

Bug #840 – Output sort order set incorrectly in CLIPROWS

In prior versions of RETL 11, the sort order in the output schema was set to be ascending, even if the incoming records were ordered descending. Records are not re-ordered by CLIPROWS, so the output schema was incorrect.

Starting in RETL 11.3, the output schema sort order is the same as the input sort order.

Bug #856 – Incorrect values are written by EXPORT if the records were previously written by EXPORT and then were transformed by BINOP

In prior versions of RETL 11, the cached byte-representation of a field set by EXPORT was not invalidated after a subsequent BINOP, resulting in invalid values written in a later EXPORT.

In RETL 11.3, the byte-representation cache is cleared in the BINOP.

Bug #886 – If “createtablemode” is “recreate” and “mode” is “truncate” in ORAWRITE, partitioning is needlessly disabled.

Setting “mode” to truncate disables partitioning because each partition would attempt to truncate the table, but specifying “recreate” truncates the table up front and makes the “mode” property meaningless.

In RETL 11.3, “mode” is set to “append” if “createtablemode” is “recreate” so that partitioning can take place if requested.

Bug #896 – EXPORT’s OUTPUT schema is incorrect

In prior versions of RETL 11, the output schema of an EXPORT (provided an OUTPUT was specified) was set to the schema defined in the “schemafile” property. This could silently drop fields that could be useful later on in the flow.

In RETL 11.3, the output schema is the same as the input schema.

Bug #1069 – REMOVEDUP does not issue a warning about unsorted data

REMOVEDUP requires sorted data to guarantee correct results. In prior versions of RETL 11, no warning was displayed if the input records were not sorted.

Starting with RETL 11.3, the following warning is displayed if the input to REMOVEDUP is not sorted:

```
WARN  - W146: The input to [removedup:1] is not sorted.  Data should
be sorted according to the proper keys or you may get unexpected
results.
```

Bug #1209 – BINOP throws an IllegalArgumentException when the destination type does not support arithmetic.

In prior versions of RETL 11, if the “dest” property specified a field whose data type did not support arithmetic, an IllegalArgumentException was thrown.

Starting in RETL 11.3, the following error message is displayed:

```
FATAL - E135: Exception in operator [binop:1]
```

```
E370: Arithmetic on xxxxxx fields is not supported.
```

However, the field type specified is incorrectly specified in the error message. This will be fixed in a future release of RETL.

Bug #1210 – A NullPointerException is thrown when the destination field of a BINOP exists but is null.

In prior versions of RETL 11, a NullPointerException was thrown when the destination field of a BINOP operator had a null value.

Starting in RETL 11.3, RETL allocates a field to hold the destination value if it does not exist.

Bug #1223 – RQST00000066282: FILTER fails with IS_NULL/IS_NOT_NULL on the left

In prior versions of RETL 11, FILTER did not correctly interpret a complex expression starting with `fieldname IS NULL` or `fieldname IS NOT NULL` but instead displayed

```
FATAL - E135: Exception in operator [filter:1]
```

```
E168: Left side must be a value.
```

In RETL 11.3, RETL correctly supports IS NULL or IS NOT NULL in complex expressions.

Bug #1244 – NullPointerException in FIELDMOD when duplicating null fields

In prior versions of RETL 11, a NullPointerException was thrown when a null field was duplicated by FIELDMOD.

Starting in RETL 11.3, a null field is duplicated as a null field.

Bug #1250 – FILTER does not detect invalid syntax in multi-part expression.

Some types of invalid FILTER expressions were not detected. In RETL 11.3, the following error message is displayed giving the invalid token and expected values:

```
FATAL - E135: Exception in operator [filter:1]
```

```
E383: Invalid syntax in "filter" property near "token" at position
6.  Expecting GE, GT, LT, LE, EQ, or NE.
```

Bug #1253 – CHANGECAPTURE does not display warning when “allvalues” and “value” properties are inconsistent.

In CHANGECAPTURE, when the “allvalues” property is “false” the “value” property is used to specify the fields to compare, RETL does not require “value” to be specified but instead compares all values, acting as if “allvalues” was set to “true” instead of “false”.

Starting in RETL 11.3, RETL displays a warning if “value” is not specified when “allvalues” is set to “false”.

```
WARN  - W149: [changecapture:1] "value" is not specified.  Comparing
all fields.
```

Bug #1259 – “changecode” field in CHANGECAPTURE is not unique for each record

In prior versions of RETL 11, instead of adding a new “changecode” field to each record, RETL shared a “changecode” field among all records with the same “changecode” value. Thus, a change to the “changecode” field was reflected in all records.

Staring with RETL 11.3, RETL correctly adds a new “changecode” field to each record.

Bug #1266 – FILTER does not validate the number of INPUTs

In prior versions of RETL 11, the number of INPUTs to FILTER was not validated, resulting in a NullPointerException or ArrayIndexOutOfBoundsException exception.

Starting in RETL 11.3, one INPUT to FILTER is enforced. If more than one input is specified, an error is displayed:

```
FATAL - E135: Exception in operator [filter:1]
```

```
E348: Operator filter:1 allows only one input.
```

Bug #1281 – RQST00000067222: Incorrect gsort.HP-UX in package

An incorrect version of the gsort utility was included in the RETL 11.2.x packages. This resulted in an error when sorting by more than 9 fields. The correct version of gsort is included in the RETL 11.3 package.

Bug #1291 – If the "reduce" property of GROUPBY is not set first, an exception is thrown without a meaningful error message

In prior versions of RETL 11, if the “reduce” property was not specified before the associated calculation properties (for example, “sum”), RETL would display an error message about setting the calculation field, but would not mention the missing “reduce” property.

Starting in RETL 11.3, RETL displays the following error message about the missing “reduce” property:

```
FATAL - E111: Exception in [groupby:1] setting property "property-
name".
```

```
E395: [groupby:1] requires the "reduce" property to be set before
any operations including "property-name"="value".
```

Bug #1309 – Missing FUNCTIONARG in CONVERT’s “make_not_nullable” causes a NullPointerException

In prior versions of RETL 11, a missing FUNCTIONARG child element when CONVERT’s CONVERTFUNCTION is “make_not_nullable” resulted in a NullPointerException being thrown.

Starting in RETL 11.3, RETL checks for the FUNCTIONARG when necessary. If it is missing, RETL displays the following error:

```
FATAL - E135: Exception in operator [convert:1]
```

```
E354: CONVERTFUNCTION: "make_not_nullable" and "make_nullable"
require a <FUNCTIONARG name="nullvalue" value="..."/> tag to be set.
```

Bug #1319 – Error in NULL field comparisons in FILTER

In prior versions of RETL 11, a null value always compared equal to a non-null field.

Starting in RETL 11.3, a null field only equals another null field.

Bug #1322 – COMPARE does not resolve output schema or records correctly

In RETL 10.x, the output schema for the COMPARE operator was the joined input schemas plus the added “result” field. In prior RETL 11.x versions, the output schema was the first input schema plus the “result” field.

Starting in RETL 11.3, the output schema calculation of the COMPARE operator matches the 10.x versions.

Bug #1337 – ArrayIndexOutOfBoundsException in CONVERT if the “sourcefield” property does not specify a valid field.

In prior versions of RETL 11, an ArrayIndexOutOfBoundsException was thrown if the source field specified in the “sourcefield” property did not specify a valid field.

In RETL 11.3, the following error is displayed:

```
FATAL - E135: Exception in operator [convert:1]
```

```
E355: Field "FIELD1 " does not exist in the input schema.
```

Bug #1338 – GROUPBY display does not show all properties when using –s SCHEMAFILE

In prior versions of RETL 11, only the last set of “reduce” and associated calculation properties was displayed when the flow was displayed using the –s SCHEMAFILE option.

Starting in RETL 11.3, all sets of “reduce” and associated calculation properties are displayed.

Bug #1353 – Invalid -s parameter does not display an error

In prior versions of RETL 11, the parameter to the –s command line switch was ignored unless it was “SCHEMAFILE”.

In RETL 11.3, “schemafile”, “SCHEMAFILE”, “none” and “NONE” are recognized. RETL will display the following error if any other value is specified:

```
"Unrecognized -s option specified: specified-value"
```

Bug #1358 – CLIPROWS “sortorder” property checking is too strict.

In prior versions of RETL 11, if the INPUT dataset to CLIPROWS is ordered on more fields than are specified in the “key” property, CLIPROWS spits out an unnecessary warning about mismatched sort keys.

Starting in RETL 11.3, RETL does not display a warning message if the records are sorted by at least the fields specified in the “keys” property.

Bug #1367 – Need to gracefully recover when hostname default throws an exception

The default value for the “hostname” property is the address of the local host. Under some configurations, this can throw an UnkownHostException.

In RETL 11.3, if the system is not able to provide the address of the local host, the “hostname” property defaults to “127.0.0.1”.

Bug #1369 – RQST00000068540: Garbage collection parameters not set correctly in retl script.

In prior versions of RETL 11, the garbage collection parameters were not being set correctly because an unset environment variable was being checked to determine the operating system type.

In RETL 11.3, the retl shell script has been fixed to set garbage collection parameters set correctly.

Bug #1395 – String += "" results in an empty string as a result in PARSER

In prior versions of RETL 11, the PARSER operator set the result of appending an empty string to an empty string.

In RETL 11.3, the result of appending an empty string in PARSER is the original string.

Bug #1399 – Error messages use “source” and “sink” instead of “INPUT” and “OUTPUT”

In RETL 11.3, all error and warning messages using “source” or “sink” when referring to datasets now use “INPUT” or “OUTPUT”.

Bug #1403 – NumberFormatException in SWITCH when an invalid OUTPUT index encountered.

In prior versions of RETL 11, if the “casevalues” property of a SWITCH operator has text that cannot be converted to a number, a NumberFormatException exception is thrown rather than a meaningful error message.

Starting in RETL 11.3, an error message is displayed:

```
FATAL - E135: Exception in operator [switch:1]
```

```
E392: Property "casevalues" must use valid OUTPUT index values.
"xxxxx=y" is not a valid context because "y" is not a valid OUTPUT
index.
```

Bug #1404 – Incorrect error message when spaces used in SWITCH's “casevalues” property

In prior versions of RETL 11, spaces before the equals sign in the “casevalues” property are considered to be part of the value to be checked, even if quotes are used.

In RETL 11.3, RETL correctly interprets the use of quotes in the “casevalues” property of SWITCH.

Bug #1405 – XML output from –s SCHEMAFILE does not show all quotes

In prior versions of RETL 11, the closing quote of the operator type did not appear in the XML output generated by the –s SCHEMAFILE option. This has been corrected in RETL 11.3.

Bug #1406 – CHANGECAPTURE does not warn if INPUTs are not sorted.

Sorted datasets are required for proper processing in CHANGECAPTURE, but there is no warning displayed if the INPUTs are not sorted.

Starting in RETL 11.3, a warning is displayed if the datasets are not sorted:

```
WARN  - W154: [changecapture:1]: Requires sorted inputs.  The input
"input.v" does not seem to be sorted.  You may get unexpected
results.
```

Bug #1407 – GROUPBY does not warn if INPUTs are not sorted and “key” is specified

Sorted input is required for proper processing in GROUPBY if the “keys” property is specified, but in prior versions of RETL 11 no warning message is displayed if the input is not sorted.

Starting in RETL 11.3, a warning is displayed if the input to GROUPBY is not sorted and the “keys” property is set:

```
WARN  - W146: The input to [groupby:1] is not sorted.  Data should
be sorted according to the proper keys or you may get unexpected
results.
```

Bug #1412 –PARSER error messages do not use log4j facility.

In prior versions of RETL 11, error and warning messages from the PARSER operator were written to stderr.

Starting in RETL 11.3, all PARSER error and warning messages are logged through the log4j facility.

Bug #1463 – No warning or error is issued when more than one FUNCTIONARG is specified in CONVERT

In prior versions of RETL 11, no error or warning message was displayed when more than one FUNCTIONARG was specified in the CONVERT “convertspec” property. The last FUNCTIONARG was used and the others ignored.

Starting in RETL 11.3, RETL will display the following warning:

```
WARN  - W153: [convert:1]: More than one "FUNCTIONARG" was
specified.  Using the last specified "FUNCTIONARG".
```

Bug #1465 – ORAWRITE problems loading calculated dfloats

In prior versions of RETL 11, the SQL*Loader data file created by ORAWRITE could contain improperly formatted dfloats if the byte-representation of the dfloat was reset by a calculation on the field. The dfloat written to the data file did not respect the maximum field length property specified to SQL*Loader through the configuration file.

In RETL 11.3, calculated dfloats are properly formatted to match the field length specified in the control file.

Bug #1480 – • Bug #1480 – SORTFUNNEL and FUNNEL can set a string field’s “maxlength” improperly.

In prior versions of RETL 11, the “maxlength” property of a string field was set to the maxlength of the field in the first INPUT to a FUNNEL or SORTFUNNEL. This discrepancy caused problems if the field had a longer maximum length in other INPUTs, including (but not limited to) producing an invalid output file from EXPORT because longer strings are not truncated to match the shorter maximum length.

In RETL 11.3, the “maxlength” property of a string is set to the maximum string length of all INPUTs.

Bug #1483 – String concatenation in BINOP does not set the maximum length correctly.

In prior versions of RETL 11, when BINOP was used to append a string onto another string, the “maxlength” property of the destination field was not checked to ensure that it was long enough to hold the resultant string.

Starting in RETL 11.3, a check on the maximum length is made. If the destination field is a pre-existing field in the schema and its “maxlength” property is not large enough for the result, it is expanded and the following warning is displayed:

```
WARN  - W155: [binop:1]: The maxlength of "field-name" expanded from
(x) to (y) due to string concatenation.
```

Changes since RETL 11.2.1

Bug #785 – Number of INPUTS to SORT not validated

In RETL versions 11.0 through 11.2.1, multiple INPUTs to SORT were allowed but only the first INPUT was used. This bug allowed silent “loss” of records.

In RETL 11.2, an error is reported:

```
E348: Operator sort:1 allows only one input.
```

A flow that ran without error prior versions will now report an error if this issue is encountered. If the records in the second (and subsequent) INPUT(s) to CLIPROWS are not needed, the flow should be changed to seed the INPUT(s) to a NOOP instead.

Bug #805 – Generator sequence attributes not fully validated

In RETL versions 11.0 through 11.2.1, no warning was displayed when the schema for a SEQUENCE field contained an invalid attribute.

In RETL 11.2.2, the following warning is displayed:

```
W144: Sequence field in operator “generator:1” contains an
unrecognized attribute “attribute”
```

Bug #808 – SQL error information not reported on some SQL errors

In RETL versions 11.0 through 11.2.1, reporting of some SQL errors did not include the error information returned by the database. For example, when a table creation failed, the following was displayed:

```
Exception in [orawrite:1] : Cannot create the table: tablename
```

In RETL 11.2.2, error information provided by the database is included in the error message:

```
Exception in [orawrite:x] : Cannot create the table: tablename

Cannot execute the query CREATE TABLE etl.bug_xxx (FIELD_A
VARCHAR(2) NOT NULL, FIELD_A VARCHAR(2) NOT NULL)

ORA-00957: duplicate column name
```

Bug #849 – Sort order and keys properties lost by LOOKUP

In RETL versions 11.0 through 11.2.1, if the first INPUT to LOOKUP was sorted, the sort order and sort keys attributes were lost. (The records retained the same order; only the attributes were incorrect.) This bug caused incorrect warnings later in the flow when subsequent operators reported that sorted input was required for correct functionality.

In RETL 11.2.2, the sort order and sort keys are copied from the first INPUT to LOOKUP.

Bug #916 – Invalid number specification ignored in BINOP, FILTER, PARSER, TERAFEREAD, SWITCH, JOIN, and CONVERT

In RETL versions 11.0 through 11.2.1, conversion of text in flow properties to RETL data types was not strictly checked. For example, if a BINOP was used to add a constant value to an int8 field and the constant was specified as “1.0”, the error converting “1.0” to an int8 would be lost and the constant would actually set to 0.

In RETL 11.2.2, invalid conversions now produce an error, such as

```
E346: Error converting “1.0” to RETL type “int8”
```

Bug #938 – Incorrect delimiter used to handle null last field in ORAWRITE

In RETL versions 11.0 through 11.2.1, if ORAWRITE's "outputdelimiter" property was specified and the last field was null, the field's delimiter ('|' by default) was used instead of the specified delimiter. The incorrect delimiter was written to the database as the value of the field instead of null.

In RETL 11.2.2, the specified delimiter is used.

Bug #1081 – Backwards compatibility bug disallowing "len" and "maxlength" attributes in schema

In RETL 11.2.1, an error was displayed if both the "len" and "maxlength" attributes were specified in a schema. This bug caused a backwards compatibility issue because some schemas specified both attributes with equal values.

In RETL 11.2.2, "len" and "maxlength" can both be specified, but the values must be the same. If the values differ, the following error is displayed:

```
E328: If both "len" and "maxlength" are specified then they must be
equal.
```

Bug #1105 – FIELDMOD duplicate does not check for pre-existing field

In RETL versions 11.0 through 11.2.1, the duplicate property did not check to see whether the new field's name matched the name of an existing field. Records would contain two fields with the same name. This bug would cause inconsistent behavior in any following operators that referred to the new or pre-exiting field.

In RETL 11.2.2, the following error message is displayed:

```
E345: Field "destination" already exists while trying to duplicate
field with "source =destination"
```

Bug #1111 – PARSER does not support floating-point constants

In RETL 11.2 and 11.2.1, the PARSER operator did not support floating-point constants. This capability was added to RETL 11.2.2.

Bug #1163 – GENERATOR cannot start a partition

In RETL 11.2 and RETL 11.2.1, the GENERATOR operator could not be partitioned when creating records. (GENERATOR could be partitioned when generating fields for pre-existing records.)

This functionality has been added to RETL 11.2.2.

Bug #1175 – SORT rejects import of dfloat or sfloat read from fixed-width file

In RETL versions 11.0 through 11.2.1, if a dfloat or an sfloat were read from a file with a fixed width schema and the width was not large enough to handle the 6 digits of precision that RETL uses by default, the record would be rejected when being re-imported from disk within the SORT operator.

For example, if a dfloat with a value of 1200.0 was read from a fixed width file and the width for the field was 6, the re-import by SORT would cause the record to be rejected because the field was written out as 1200.000000, which has a length longer than 6.

In RETL 11.2.2, dfloats and sfloats are written to disk respecting the maximum length specified in the import file.

Bug #1177 – FIELDMOD does not perform deep enough copy when duplicating a field

In RETL versions 11.0 through 11.2.1, a reference to a field was copied when the duplicate property was specified. Any change to one field would also show up in the other field.

In RETL 11.2.2, a new field is constructed and its value set to the same as the source field.

Bug #1183 – Incorrect error message when FILTER's filter property syntax is incorrect

In RETL versions 11.0 through 11.2.1, RETL displayed the following error message when the syntax of the filter property of FILTER was incorrect:

No filter logic is specified.

In RETL 11.2.2, if the filter property syntax is incorrect, the following error is displayed:

```
E349: Invalid syntax in “filter” property: filter-value
```

Bug #1211 – 32bit JVM is not properly installed on AIX

In RETL 11.2.1, the tarball for the 32-bit JVM for AIX was corrupt. This bug has been fixed in RETL 11.2.2.

Bug #1222 – NullPointerException in TERAWRITE

In RETL 11.2 and 11.2.1, if TERAWRITE was used to write data to a Teradata table that had more columns than the dataset and in a different order, it is possible for TERAWRITE to throw a NullPointerException because the dataset fields were not correctly mapped to the table columns.

This bug has been corrected in RETL 11.2.2.

Bug #1226 – Incorrect handling of DECIMAL fields in TERAFEREAD

In RETL 11.2 and 11.2.1, TERAFEREAD incorrectly read DECIMAL fields from the database.

This bug has been corrected in RETL 11.2.2.

Bug #1237 – RQST00000061454: orawrite strips trailing spaces breaking CHANGECAPTURE

In RETL versions 11.0 through 11.2.1, spaces at the end of records are not preserved in ORAWRITE because the default behavior of SQL*Loader is to trim the trailing spaces. This bug can cause problems when the data is later retrieved and compared against another dataset using CHANGECAPTURE because the trimmed trailing space causes the record to be flagged as a change.

RETL 11.2.2's ORAWRITE operator accepts a new property “controlfileloadoptions” that will be written to the SQL*Loader control file. Setting “controlfileloadoptions” to “PRESERVE BLANKS” will preserve trailing spaces. This action must be coupled with a delimited load (specified by setting the “outputdelimiter” property) in order to work well with CHANGECAPTURE.

Changes since RETL 11.2

See above for enhancements in the 11.2.1 release.

- Fixes Bug#434 – Allow logging to log actual times spent per operator.
- Fixes Bug#544 – Missing CONVERTFUNCTION results in no-op instead of error.
- Fixes Bug#860 – 4 RETL instances much quicker than 4 ORAREADs.
- Fixes Bug#905 – Progress indicators.
- Fixes Bug#906 – Performance statistics show operator initialization time under framework.
- Fixes Bug#934 – DBLOOKUP does not partition.
- Fixes Bug#936 – Partitioning error when funneling multiple partition groups with different numpartitions.
- Fixes Bug#942 – Incorrect schema resolution in generator after inline funnel.
- Fixes Bug#945 – Negative dfloats can break fixed-length output schema.
- Fixes Bug#946 – Occasional NullPointerException in funnel.
- Fixes Bug#953 – Join key datatypes must match.
- Fixes Bug#955 – Enhanced logging
- Fixes Bug#957 – Assertions should be enabled when RFX_DEBUG=1.
- Fixes Bug#961 – Partitioned lookup is incorrect when lookup table is non-keyed partitioned.
- Fixes Bug#963 – Mismatched numpartition settings cause assertion error.
- Fixes Bug#964 – Date comparison failure in parser.
- Fixes Bug#966 – Should be able to specify multiple outputs in parser.
- Fixes Bug#967 – IndexOutOfBoundsException in orawrite.
- Fixes Bug#969 – Groupby errors on null value for min/max.
- Fixes Bug#970 – Double formatting with magnitude < 0.
- Fixes Bug#971 – first field nulls not handled properly in comparison and assignments.
- Fixes Bug#972 – Enhance graphing with performance measurements.
- Fixes Bug#973 – Verify graphing performance measurement accuracy.
- Fixes Bug#974 – Add and adjust dataset monitoring capabilities.
- Fixes Bug#975 – Improve the tests surrounding the new parser functionality.
- Fixes Bug#981 – Partitioned lookup may not scale.
- Fixes Bug#982 – Delimited orawrite should specify length for CHAR.
- Fixes Bug#984 – DEFAULTs in rfx.conf do not override operator properties.
- Fixes Bug#985 – NullPointerException from StreamRedirector.
- Fixes Bug#986 – Comparison between different field types gives incorrect error.

- Fixes Bug#990 – DBLOOKUP sets incorrect SQL to look up in database.
- Fixes Bug#994 – -g must be last option on command line.
- Fixes Bug#995 – Convert can invalidate sort order on presorted data.

Changes since RETL 11.1

- Fixes Bug#407 – In framework we tread all int and uint type as long in Java version.
- Fixes Bug#421 – Rejected records get printed to the screen which can produce too much output.
- Fixes Bug#441 – Support for properly-formatted precision in dfloat/sfloat.
- Fixes Bug#445 – Parallel file reads.
- Fixes Bug#461 – Include specific error in import rejects.
- Fixes Bug#464 – Determine if we really need hostname/port for ORAREAD/ORAWRITE.
- Fixes Bug#484 – Need build information option.
- Fixes Bug#488 – Better exception handling in Operator base class.
- Fixes Bug#493 – Record key hash non-unique with multiple key fields.
- Fixes Bug#499 – Import does not use nullvalue property to detect null values.
- Fixes Bug#503 – Fixed width of zero not allowed.
- Fixes Bug#507 – Schema.parseKeyFields(...) should be more robust.
- Fixes Bug#508 – Number format is not standardized on export.
- Fixes Bug#510 – userid/password properties do not exist for db2read/teraferead operators.
- Fixes Bug#515 – JDBCWRITE should set proper type in constructor.
- Fixes Bug#516 – Implement native load mechanisms for orawrite in 11.x versions.
- Fixes Bug#517 – Implement native load mechanisms for terawrite in 11.x versions.
- Fixes Bug#522 – Generator does not implement partnum_offset or partcount_incr.
- Fixes Bug#532 – final_delimiter not implemented.
- Fixes Bug#534 – Missing len from fixed schema file causes delimited export.
- Fixes Bug#541 – Incorrect handling of too few fields in import.
- Fixes Bug#543 – Multiple use of same dataset as input throws null pointer exception.
- Fixes Bug#547 – JDBCWRITE should check schema owner.
- Fixes Bug#548 – Port is not configurable for Teradata (terawrite).
- Fixes Bug#553 – Checks if there is sqlldr.
- Fixes Bug#555 – Strip schema attribute not handled.
- Fixes Bug#557 – fieldmod does not check if fieldnames actually exist in incoming dataset.
- Fixes Bug#558 – createtablemode should only allow recreate/create.

- Fixes Bug#561 – RETL does not abort gracefully if dbwrite password not specified.
- Fixes Bug#564 – Should modify maxsize if database column maxsize < field maxsize.
- Fixes Bug#572 – Merge jdbcread methods with DatabaseUtils and ColumnInfo.
- Fixes Bug#574 – logical error in CHANGECAPTURE.
- Fixes Bug#579 – Fatal error trapping when using SQL-loader.
- Fixes Bug#606 – Handling environment problems with RETL executable/library.
- Fixes Bug#610 – In the fixed import, we should check if the total length of the record equals the sum of the length of fields.
- Fixes Bug#632 – Incorrect handling of too many fields in import.
- Fixes Bug#635 – Generator does not create null values same as 10.x.
- Fixes Bug#641 – Generator sequence does not respect destination field datatype.
- Fixes Bug#643 – Convert cannot handle duplicate source fields in one CONVERTSPEC.
- Fixes Bug#647 – Reject file incorrect format.
- Fixes Bug#648 – Incorrect contents of reject file.
- Fixes Bug#675 – Sort of generated string fails on re-import.
- Fixes Bug#678 – Multibyte fixed width export incorrect/slow.
- Fixes Bug#685 – Funnel should not be waiting when there is no data in any one of inputs.
- Fixes Bug#686 – Dataset should check with deadlock listener always even though the dataset has been expanded.
- Fixes Bug#694 – Non-UTF encodings cause exception in import.
- Fixes Bug#696 – 11.x can break the output schemafile definition.
- Fixes Bug#698 – sortfunnel outputs wrong data – logical problem.
- Fixes Bug#700 – Generator sequence does not work with doubles.
- Fixes Bug#701 – Binop incorrect when destination field is a source field.
- Fixes Bug#703 – Improved performance when handling errors.
- Fixes Bug#705 – Performance bottlenecks in orawrite.getRecordStringBuffer(...).
- Fixes Bug#708 – Performance improvement in jdbcread.
- Fixes Bug#709 – Performance degradation in jdbcread due to string comparison.
- Fixes Bug#710 – Sort does not handle non-existent 'key' gracefully.
- Fixes Bug#713 – CHANGECAPTURE codes incorrect in documentation.
- Fixes Bug#714 – Error message bad when null value is in non-null field.
- Fixes Bug#716 – Delimiter property of sort not working correctly.
- Fixes Bug#718 – NullPointerException in lookup.
- Fixes Bug#719 – RIGHTOUTERJOIN invalid schema.

- Fixes Bug#721 – Import of multi-byte fixed-width files fails.
- Fixes Bug#722 – ConvertByteToBytes incorrect.
- Fixes Bug#723 – LONGVARCHAR only partially supported.
- Fixes Bug#724 – Cannot handle "select NULL something from test_jdbc".
- Fixes Bug#725 – Generator backwards compatibility issue with field length.
- Fixes Bug#726 – ArrayIndexOutOfBoundsException when missing field type in schema.
- Fixes Bug#727 – NullPointerException when numloader > 1 and RFX_SHOW_SQL=1.
- Fixes Bug#728 – I18N and fixed width incorrect.
- Fixes Bug#729 – The dataset hang when GENERATOR->NOOP with independent thread with very small buffersize and huge data.
- Fixes Bug#734 – NullPointerException when paging.
- Fixes Bug#736 – ArrayOutOfBounds in export.
- Fixes Bug#737 – Rejects file not flushed when exiting due to too many rejects.
- Fixes Bug#738 – Parallel orawrite does not alternate TEMPDIRs.
- Fixes Bug#739 – tempdir property should override setting in rfx.conf.
- Fixes Bug#740 – intra-operator parallelism properties should have standard names.
- Fixes Bug#743 – Temporary directories should be checked.
- Fixes Bug#745 – Schema assembler should ensure field delimiter != record delimiter.
- Fixes Bug#746 – orawrite should ensure output delimiter != record delimiter.
- Fixes Bug#747 – orawrite cannot handle preload and postload for stored procedure.
- Fixes Bug#748 – Convoluted error output when setting properties.
- Fixes Bug#749 – Issue warning instead of error for unrecognized properties.
- Fixes Bug#751 – Combine DataSet, SDataSet, DataSet_Simple, and DataSet_RawBytes.
- Fixes Bug#752 – Add --graphviz-files
- Fixes Bug#753 – sp_prequery should be executed BEFORE the query.
- Fixes Bug#754 – Heterogeneous output schemas.
- Fixes Bug#756 – Add 64bit JVM support in RETL 11.
- Fixes Bug#757 – Convert does not handle TYPEPROPERTY properly.
- Fixes Bug#758 – Int32/Int64 Field type rollover.
- Fixes Bug#760 – Binop odd division rounding error.
- Fixes Bug#761 – Lookup returns false matches.
- Fixes Bug#762 – Unsigned integer issues.
- Fixes Bug#764 – NumberUtils.ConvertIntToBytes poor algorithm.

- Fixes Bug#769 – Sort failure on date without century.
- Fixes Bug#771 – Sort is not removing temporary files.
- Fixes Bug#772 – RETL 11.x does not support partitioning.
- Fixes Bug#774 – Lookup ArrayIndexOutOfBoundsException with multiple lookup tables.
- Fixes Bug#776 – Out of memory exception should abort flow.
- Fixes Bug#779 – Fieldmod loses SortKeys.
- Fixes Bug#780 – Schema assembler sets sort keys incorrectly.
- Fixes Bug#782 – Default value for properties not set for join.
- Fixes Bug#784 – graphviz cleanup.
- Fixes Bug#786 – Should error when record len != sum(field lens) in schemafile.
- Fixes Bug#788 – Forwards compatibility for operator properties.
- Fixes Bug#790 – Join with database operator.
- Fixes Bug#791 – Groupby requires reduce property before Min, Max ... Fields.
- Fixes Bug#792 – Verify retl.jar during RETL startup.
- Fixes Bug#793 – Verify retl.jar during RETL startup.
- Fixes Bug#796 – Enhanced performance logging.
- Fixes Bug#799 – Deadlock detection not working for pipelines with more than two outputs.
- Fixes Bug#800 – Occasional errors when paging.
- Fixes Bug#801 – Paging may result in too many files open.
- Fixes Bug#802 – DataSet paging files may not be created.
- Fixes Bug#803 – Unneeded check for duplicate tables in existence.
- Fixes Bug#811 – LEFTOTERJOIN does not support spaces in nullvalue property.
- Fixes Bug#812 – fieldmod blows away sorted order.
- Fixes Bug#813 – RIGHTOUTERJOIN invalid sort order.
- Fixes Bug#816 – Record.Join uses incorrect calculation to get # of fields.
- Fixes Bug#818 – Poor message displayed when output record breaks schema.
- Fixes Bug#821 – Delimited import strips trailing whitespace in 11 but not in 10.
- Fixes Bug#822 – Package debug/non-optimized version of retl.jar
- Fixes Bug#823 – The RETL Programmer's Guide description of ifnotfound property is incorrect.
- Fixes Bug#824 – Improve logging information with flow name.
- Fixes Bug#825 – Paths in RETL script create problems on Windows.
- Fixes Bug#826 – Add 'loader_options' parameter to orawrite.

- Fixes Bug#829 – Maxlength of 0 creates problems for orawrite.
- Fixes Bug#830 – slsildmdm runs faster in 10.x than in 11.x.
- Fixes Bug#832 – Incorrect null in FULLOUTERJOIN.
- Fixes Bug#833 – Error in double to byte.
- Fixes Bug#834 – dotty files removed with -g option.
- Fixes Bug#835 – Ned specific error message in ORAREAD query.
- Fixes Bug#836 – Missed record in multi-threaded import.
- Fixes Bug#838 – Stack overflow in generator if limit < init.
- Fixes Bug#839 – Copy does not perform deep enough copy.
- Fixes Bug#841 – dfloat fields can break fixed output schema.
- Fixes Bug#842 – Lookup table loading is dependant on data record production.
- Fixes Bug#843 – Assertion is fired in allocation size_profile script.
- Fixes Bug#845 – Assertion in Parser_ByteSequence.readFixedField in Allocation size_profile.
- Fixes Bug#847 – Pipeline initialization and schema resolution is serial.
- Fixes Bug#848 – ArrayIndexOutOfBoundsException when 1 INPUT provided to lookup.
- Fixes Bug#852 – Operator JDBCWRITE needs parallel property.
- Fixes Bug#853 – Check for WHERE in DBLOOKUP select_sql property incorrect.
- Fixes Bug#854 – Partitioned Orawrite runs preload and postload for each partition.
- Fixes Bug#855 – db2write deletes log files even after a fatal error.
- Fixes Bug#859 – Monitor method of funnel does not work for empty datasets.
- Fixes Bug#860 – 4 RETL instances much quicker than 4 ORAREADs.
- Fixes Bug#862 – Adding batched funnel algorithms causes dataset._pblock to be null.
- Fixes Bug#864 – Should print a warning if orawrite cannot be partitioned.
- Fixes Bug#867 – verboserejects should be a visible property in import.
- Fixes Bug#868 – RETL does not print any warning if rejects + verboserejects=false.
- Fixes Bug#869 – Errors not caught in DatabaseUtils.Connect().
- Fixes Bug#871 – ImportRejectSupporter throws an error on processing rejected records.
- Fixes Bug#872 – Multiple-query JDBCRead is slower than multiple JDBCReads.
- Fixes Bug#873 – jdbcread needs to partition.
- Fixes Bug#874 – Import partitioning does not calculate number of partitions correctly.
- Fixes Bug#875 – Import partitioning does not need to end thread.
- Fixes Bug#877 – verify_retl chokes on space between -d and database type.

- Fixes Bug#881 – Schema compatibility checking in export is too strict.
- Fixes Bug#883 – Partitioning fails in aipt_future_delivery.
- Fixes Bug#887 – Execute jdbcread subthreads in parallel.
- Fixes Bug#888 – Dataset-less funnel.
- Fixes Bug#889 – Check for retl.jar in RETL script does not check RETL_DIST_LIB variable.
- Fixes Bug#890 – Funnel should have 11.1 algorithms in 11.2.
- Fixes Bug#891 – Generated HTML for graphs contains incorrect message about dotty file.
- Fixes Bug#892 – Funnel should drop sort order.
- Fixes Bug#895 – Add create_options property to orawrite.
- Fixes Bug#901 – Include 1.4.1 AIX JRE in package rather than 1.4.0.
- Fixes Bug#903 – Inlining lookup may improve performance.
- Fixes Bug#904 – Splitter needs to end thread when partitioning.
- Fixes Bug#907 – Non-deterministic output if debugging is set on.
- Fixes Bug#910 – Incorrect schema resolution for fullouterjoin.
- Fixes Bug#911 – Binop produces incorrect field type when multiplying int by dfloat.
- Fixes Bug#912 – NullPointerException in binop when left or constleft not specified.
- Fixes Bug#913 – Conflicting versions of rfx.conf num partitions and -n option do not seem to resolve properly.
- Fixes Bug#919 – Import partitioning does not work with multiple files.
- Fixes Bug#920 – Poor error message when nullable missing in import schemafile.
- Fixes Bug#921 – Hash partitioning after splitter uses incorrect dimensions.
- Fixes Bug#922 – Import - multi-threaded imports are losing records.
- Fixes Bug#923 – Export should partition.
- Fixes Bug#924 – FileUtils concatenation() errors when output file does not exist yet.
- Fixes Bug#925 – Misleading error message "record was too long" in import.
- Fixes Bug#927 – Changecapture algorithm does not allow for partitioned imports.
- Fixes Bug#929 – install_jre fails on some servers.
- Fixes Bug#930 – Cannot override default logger file from config file.
- Fixes Bug#931 – Log file should be in temp directory if directory not specified.
- Fixes Bug#932 – Unable to write to log file.
- Fixes Bug#935 – CHANGECAPTURELOOKUP interferes with LOOKUP's tablekeys property.