

Retek[®] Extract Transform and Load[™] 11.2.2

Notes de mise à jour

Siège social :

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
Etats-Unis

888.61.RETEK (numéro vert
aux Etats-Unis)
Standard :
+1 612 587 5000

Télécopie :
+1 612 587 5100

Siège européen :

Retek
110 Wigmore Street
London
W1U 3RW
Royaume-Uni

Standard :
+44 (0)20 7563 4600

Département commercial :
+44 (0)20 7563 46 46

Télécopie:
+44 (0)20 7563 46 10

Le logiciel décrit dans la présente documentation fait l'objet d'un accord de licence. Il se compose d'informations confidentielles qui restent la propriété de Retek Inc. et son utilisation est soumise au respect des dispositions de cet accord.

Aucune partie de cette documentation ne peut être reproduite ni transmise sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite expresse de Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403, USA. Il est interdit de supprimer la notice de copyright sans l'autorisation de Retek Inc.

Les informations contenues dans cette documentation sont susceptibles d'être modifiées sans préavis.

Retek propose la documentation relative à ses produits en lecture seule afin d'assurer l'intégrité de son contenu. Le support clientèle Retek ne peut pas prendre en charge toute documentation modifiée sans l'autorisation de Retek.

La fonction ici décrite s'applique à cette version, telle que mentionnée en titre de ce document, et à nulle autre version logicielle, y compris et sans se limiter aux mises à jour postérieures dudit composant logiciel. La fonction ici décrite est susceptible d'être modifiée avec les mises à jour des nouvelles versions logicielles et Retek se réserve le droit d'apporter toute modification à son entière discrétion.

Retek[®] Extract Transform and Load[™] est une marque de Retek Inc.

Retek et le logo Retek sont des marques déposées de Retek Inc.

Ce document non publié est protégé par un accord de confidentialité et par la législation sur le secret commercial, les droits d'auteur, etc. En cas de publication, la mention suivante s'appliquera :

©2005 Retek Inc. Tous droits réservés.

Support clientèle

Horaires du support clientèle

Le support clientèle est disponible 7 jours sur 7, 24 heures sur 24 et 365 jours par an, par courrier électronique, téléphone et Internet.

Selon l'option de support choisie par chaque client (Standard, Plus ou Premium), les horaires de certains services peuvent être restreints. Les problèmes critiques (gravité 1) sont pris en charge 7 jours sur 7, 24 heures sur 24, jusqu'à leur résolution, pour tous les clients bénéficiant d'un contrat en cours de validité. Les clients de Retek bénéficiant d'un contrat en cours de validité peuvent contacter, selon les termes du contrat, un représentant du support clientèle global d'une des manières suivantes.

Méthode de contact

Coordonnées

Courrier électronique

support@retек.com

Internet (ROCS)

rocs.retek.com

consultation des problèmes

Site Web client sécurisé de Retek pour la mise à jour et la

Téléphone

+1 612 587 5800

Des numéros verts sont également disponibles dans différentes régions du monde :

Australie	+1 800 555 923 (AU-Telstra) ou +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Corée	00 308 13 1342
Royaume-Uni	0800 917 2863
Etats-Unis	+1 800 61 RETEK ou 800 617 3835

Adresse postale

Retek Customer Support
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403

Lorsque vous contactez le support clientèle, veuillez fournir :

- la version du produit et le nom du programme/module,
- la description fonctionnelle et technique du problème (y compris l'impact commercial),
- les instructions détaillées étape par étape pour reproduire le problème,
- l'intitulé exact du message d'erreur reçu,
- une capture d'écran de chaque opération effectuée.

Présentation

Cette mise à jour RETL 11.2.2 est une mise à jour de correctif contenant uniquement des correctifs d'erreurs. Les erreurs qui ont été corrigées sont expliquées dans la section « Modifications depuis la mise à jour 11.2.1 » ci-dessous.



Remarque : Plusieurs corrections d'erreurs permettent d'intercepter des erreurs dans des flux qui n'étaient pas détectées dans des versions antérieures de RETL. Avant de passer à RETL 11.2.2, Retek vous conseille de réaliser un test de régression complet en prêtant une attention toute particulière à l'erreur n°916.

Prise en charge de bases de données

La version de RETL 11.2.2 apporte les modifications de prise en charge des bases de données de RETL suivantes :

- Teradata 2 version 4.1 n'est plus prise en charge par RETL. Elle a été remplacée par la prise en charge de Teradata 2 Release 5.0.
- DB2 Version 7 n'est plus prise en charge par RETL. La prise en charge de DB2 Version 8.1 demeure valide.

Reportez-vous à la section « Matrice de compatibilité » ci-dessous pour connaître la liste des plates-formes et des bases de données prises en charge.

Certification du produit

RETL 11.2.2 est certifié RDW 10.2.3. Reportez-vous à la section « Certification » ci-dessous pour obtenir la matrice des produits Retek ainsi que la version certifiée de RETL.

Avant de procéder aux travaux d'installation ou de développement, Retek vous recommande vivement de lire l'intégralité du Guide du programmeur RETL 11.2.2 fourni avec la mise à jour. Il est particulièrement important de lire et comprendre le « Chapitre 2 – Installation et configuration du système ». En outre, soyez très attentif à la section « Problèmes connus » des notes de mise à jour.

Les groupes de produits doivent répondre à l'ensemble des certifications et des tests régression/volume de la version RETL 11 avant que le déploiement ne soit réalisé au niveau des sites clients.

Le numéro de version de RETL 11.2.2 est 558. Une fois l'installation de RETL 11.2.2 réussie, l'exécution de RETL avec l'option de ligne de commande `-v` doit faire apparaître ceci :

```
$ retl -v  
retl 11.2.2 build 558
```

Les clients disposant de la mise à jour RETL 11.2.1, de mises à jour antérieures et/ou de mises à jour de correctifs postérieurs sont vivement encouragés à procéder à la mise à niveau vers RETL 11.2.2 une fois le test de régression complet réalisé. Les problèmes identifiés dans les mises à jour RETL 11 et les mises à jour antérieures seront résolus par le biais d'optimisations et de correctifs, délivrés via les mises à jour de correctifs RETL 11. Aucune autre mise à jour n'est prévue pour RETL 10.x.

Optimisations de RETL 11.x

- Mise en application du partitionnement/redimensionnement

Le concept de partitionnement doit être mis en parallèle autant que possible de manière à entraîner, dans l'idéal, un flux non ralenti par les goulots d'étranglement dans RETL. L'objectif est de coder les flux de manière à ce qu'ils puissent être partitionnés afin d'être exécutés dans une fenêtre de batch du client.
- Principales optimisations des performances
 - Les mises à jour RETL 11 incluent une nouvelle architecture qui optimise le parallélisme pipeline en rassemblant, si possible, les opérateurs connectés dans le même pipeline. Cette architecture permet de réduire le nombre de traitements requis et ainsi de minimiser la commutation et l'emballage du contexte parmi les traitements. Le cadre prend ainsi en charge des performances plus élevées.
 - Un algorithme en entonnoir optimisé permet d'augmenter de manière importante la vitesse des flux contenant l'opérateur en entonnoir.
 - Les types internes sont désormais représentés sous la forme d'octets natifs. La conversion des chaînes de caractères n'étant plus nécessaire, il est possible d'économiser de l'espace et d'augmenter le niveau de performances.
 - Les performances dans le cadre du traitement des objets rejetés par l'opérateur d'importation sont optimisées. Les utilisateurs peuvent désormais désactiver la « prolixité » des rejets d'impression à l'écran. Il est ainsi possible d'obtenir des améliorations importantes au niveau des modules RETL qui rejettent les enregistrements lors de l'importation.
 - Les performances au niveau de la lecture et de la mise en application des ensembles de données qui connectent les opérateurs ont été optimisées de manière importante.
- Traitement des erreurs
 - Les mises à jour RETL 11 offrent un meilleur débogage des flux en affichant des messages d'erreur spécifique aux lignes et aux colonnes lorsque l'application RETL rencontre des problèmes pour lire une définition de flux XML. Le traitement des exceptions de RETL 11 permet également de procéder à un débogage et à un traitement des erreurs de meilleure qualité.
 - Compte tenu des possibilités de configuration du seuil d'erreur en écriture, RETL peut être configuré de manière à annuler l'opération lorsque le nombre maximal d'enregistrements rejetés par SQL*loader est atteint.
 - La qualité générale des messages d'erreur a été améliorée dans la version 11.1 par rapport aux versions 10.x.

- Débogage et enregistrement
 - Graphiques de flux visuel RETL
A compter de l'application RETL 11.2, il est possible d'utiliser une option de ligne de commande pour générer le graphique visuel d'un flux RETL, incluant les noms des opérateurs, les noms des liens et les désignations des traitements. Pour plus d'informations, reportez-vous au Guide du programmeur RETL.
 - Meilleures performances d'enregistrement
RETL 11.2 enregistrera désormais les événements importants, tels que l'exécution de requêtes, le classement via gsort, le chargement de données via des utilitaires de chargement de la base de données natifs (SQL*loader, par exemple), etc.
 - Graphique points principaux/goulots d'étranglement des performances
RETL exporte désormais un graphique HTML des performances. Ce graphique utilise des indicateurs visuels pour mettre en évidence les goulots d'étranglement du flux de manière à ce que les développeurs de flux, les analystes de performances et les conseillers en matière de services puissent identifier la cause à l'origine des flux aux faibles performances.
 - RETL 11 utilise l'utilitaire de journalisation log4j pour tous les résultats d'enregistrement des performances et du terminal. L'utilitaire log4j est un produit libre extrêmement flexible géré par la fondation Apache Software Foundation. Pour obtenir des informations complémentaires sur log4j, visitez le site <http://logging.apache.org/log4j/docs/documentation.html> et reportez-vous au Guide du programmeur RETL..
- Nouveaux opérateurs
 - Opérateur PARSE
La logique commerciale peut être codée dans un script de code similaire à Java qui permet d'éviter la prolifération des opérateurs, d'améliorer les performances en réduisant le nombre d'opérateurs (FIELDMOD, BINOP, FILTER, SWITCH, COPY, par exemple), de réduire les coûts de maintenance en diminuant la complexité des flux et de contribuer à une meilleure définition des tailles lors du partitionnement des flux.
 - Opérateur DBLOOKUP
Cet opérateur permet d'associer directement le flux aux enregistrements dans une table de la base de données.
 - Opérateur CHANGECAPTURELOOKUP
Fonctionnement similaire à celui de l'opérateur changecapture, il accepte néanmoins des entrées non classées qui permettent un partitionnement et une définition des tailles de meilleure qualité.
 - Opérateur SPLITTER
Un opérateur de partitionnement, basé sur une chaîne d'événements intervenant en séquence périodiquement, qui permet de distribuer les enregistrements de manière séquentielle au niveau des différentes partitions.
- Documentation
 - Guide du programmeur, version améliorée de manière importante
 - Guide de syntonisation des performances
 - Guide des pratiques d'excellence

- **Aide en ligne**
Une option de ligne de commande a été ajoutée aux mises à jour RETL 11 de manière à ce que les développeurs de flux puissent visualiser l'utilisation et la syntaxe des opérateurs via la ligne de commande sans avoir à se reporter au présent document.
- **Installation et configuration simplifiées**
Les mises à jour RETL 11 regroupent le code dans un seul code binaire (il existait 26 codes binaires dans la mise à jour 10.x). L'installation se trouve donc grandement facilitée. La configuration de l'environnement nécessaire à l'installation est en outre moins importante.
- **Réelle indépendance par rapport aux plates-formes**
La nouvelle architecture des mises à jour RETL 11 a entièrement été rédigée dans Java. Par conséquent, il est désormais possible d'exécuter RETL sur des plates-formes qui nécessitaient autrefois d'importants efforts dans le cadre du portage. Pour connaître les plates-formes prises en charge, reportez-vous au Guide du programmeur et à la matrice de compatibilité. Une des conséquences directes de l'indépendance par rapport aux plates-formes est que d'importantes ressources de RETL nécessaires à la maintenance de la plate-forme sont désormais libérées et permettent le développement de nouvelles fonctions modernes.
- **Réelle indépendance par rapport aux bases de données**
Les mises à jour RETL 11 disposent d'opérateurs « dbread »/« dbwrite » génériques. Ils utilisent une technologie JDBC pour se connecter aux bases de données de manière totalement indépendante. Les mises à jour 10.x antérieures étaient incapables de se connecter à différents types de bases de données dans un même flux. Les mises à jour RETL 11 permettent de passer d'une base de données à l'autre dans un même flux (d'une base de données DB2, Oracle ou Teradata à une autre dans un même flux, par exemple). Pour plus d'informations à ce sujet, reportez-vous au Guide du programmeur RETL.
- **Compatibilité ascendante avec les mises à jour antérieures**
Les versions de l'application RETL 11 doivent être entièrement compatibles avec les versions 10.x. Les versions de l'application RETL 11 sont néanmoins plus strictes en ce qui concerne la mise en application d'interfaces de flux XML valides, les schémas de saisie et les champs de données. Il peut arriver que le produit RETL 11 signale des erreurs au niveau des données et/ou du flux qui n'auraient pas été détectées auparavant. Il est donc possible que le nombre d'enregistrements supprimés ou annulés et/ou de messages d'erreur soit plus important qu'auparavant.

Matrice de compatibilité

Le tableau suivant représente les combinaisons Base de données/SE qui sont prises en charge et qui disposent d'une certification assurant un fonctionnement correct.

SE	Version	Arch	BD	Version
AIX	5.1	32/64	none	none
AIX	5.1	64	Oracle	9.2
AIX	5.1	64	DB2	8.1.3
AIX	5.1	64	Teradata	2r5.0
AIX	5.2	32/64	none	none

SE	Version	Arch	BD	Version
AIX	5.2	64	Oracle	9.2
AIX	5.2	64	DB2	8.1.3
AIX	5.2	64	Teradata	2r5.0
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	11.11	64	Teradata	2r5.0
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
Solaris	9	32/64	Teradata	2r5.0
Solaris	9	64	DB2	8.1.3



Remarque : RETL 11 requiert uniquement un programme binaire et une installation pour l'ensemble des combinaisons plate-forme/base de données répertoriées ci-dessus.

Certification

Le tableau suivant répertorie les produits dont le fonctionnement avec une version spécifique de RETL est certifié.

Produit	Version de RETL
RMS 11.0.2	RETL 11.2.1
RPM 11.0.1	RETL 11.2.1
ReIM 11.0.2	RETL 11.2.1

Produit	Version de RETL
RDW 11.0	RETL 11.2.1
RDW 10.2.3	RETL 11.2.2

Installation

Reportez-vous au Guide du programmeur RETL pour connaître les instructions d'installation et de configuration.

Contenu du produit

Voici une brève description du contenu de retl.11.2.2.tar.gz :

bin/

Comprend des fichiers exécutables requis pour l'exécution de RETL sur différentes plates-formes.

Fichier	Description
gsort.SunOS	gsort pour SunOS
gsort.HP-UX	gsort pour HP-UX
gsort.AIX	gsort pour AIX
retl	Fichier exécutable RETL
rfx	Lien symbolique vers retl fourni pour la compatibilité ascendante
verify_retl	Outil de vérification RETL
README.verify_retl	README pour verify_retl
install_graphing.ksh	Installe le module graphique GraphViz
install_jre.ksh	Installe l'environnement d'exécution Java
test_funcs.ksh	Utilisé par verify_retl pour exécuter des flux test



Remarque : Un seul programme binaire (retl/rfx) est requis pour l'ensemble des combinaisons base de données et plate-forme

lib/

Il s'agit de fichiers jar nécessaires pour exécuter RETL.

Fichier	Description
retl.jar	Bibliothèque d'exécution RETL

Fichier	Description
db2java.zip	Pilote JDBC DB2
log4j-1.2.9.jar	Utilitaire d'enregistrement log4j
ojdbc14.jar	Pilote JDBC Oracle
teradata.jar	Pilote JDBC NCR Teradata
xercesImpl.jar	Analyseur Xerces XML
xml-apis.jar	XML APIs
xmlParserAPIs.jar	XML APIs
tools.jar	Optimisations futures
activation.jar	Optimisations futures

etc/

Ce répertoire contient la documentation associée à RETL.

Fichier	Description
rfx.conf	Fichier de configuration par défaut de RETL.
logger.conf	Fichier de configuration de journalisation par défaut.

samples/

Ce répertoire contient les exemples fournis avec RETL. Reportez-vous au fichier README dans le répertoire samples/ pour obtenir des informations complémentaires sur les exemples fournis avec RETL. Il s'agit des exemples exécutés après l'installation, par `verify_retl`

graphing/

Ce répertoire contient la suite d'applications Graphviz permettant la visualisation graphique des flux RETL. Reportez-vous à la section « Produire des sorties graphiques de flux avec RETL » du Guide du programmeur RETL.

JRE/

Ce répertoire contient l'environnement d'exécution JRE (Java Runtime Environment) requis par RETL.

Problèmes connus

Principaux problèmes connus

- L'application RETL ne prend pas en charge les mathématiques arbitraires de précision.

- Par défaut, l'opérateur d'exportation utilisera le délimiteur de champs « | » si la propriété « schemafile » n'est pas définie. Une telle opération risque d'entraîner des erreurs si les données entrantes contiennent le caractère « | ». S'il est possible que les données contiennent le caractère « | », nous vous recommandons d'utiliser un fichier de schéma d'exportation.
- Les opérateurs joints peuvent utiliser une quantité importante de mémoire lorsque la distribution des enregistrements de même importance est faible et que le nombre d'enregistrements est élevé. Il est important de vérifier les opérations jointes à l'aide de volumes de données adaptés au niveau de chaque entrée de manière à connaître la quantité maximale de mémoire qui peut être nécessaire à votre application.

Différences entre les versions 10.x et 11.x de l'application RETL

- Les versions de l'application RETL 11 sont plus strictes en ce qui concerne la mise en application d'interfaces de flux XML valides, les schémas de saisie et les champs de données. Il peut arriver que le produit RETL 11 signale des erreurs au niveau des données et/ou du flux qui n'auraient pas été détectées auparavant. Il est donc possible que le nombre d'enregistrements supprimés ou annulés et/ou de messages d'erreur soit plus important qu'auparavant.
- Une fonction non intentionnelle de la version 10.x permettait aux fichiers de données de saisie de contenir un nombre de champs supérieur à celui défini dans le fichier de schéma. Dans RETL 11, le fichier de données ne peut pas contenir plus de champs. Si c'est le cas, une exception est générée. Cette dernière est enregistrée sous le nom de bug#632 et sera rectifiée dans une version ultérieure de RETL 11.
- Il est possible que l'ordre des champs de sortie des versions 10.x et 11.x diffère pour les flux dont la propriété « schemafile » n'est pas définie au niveau de l'opérateur d'exportation.
- Les versions 11.x de l'application RETL nécessitent des propriétés supplémentaires pour les opérateurs « dbread »/« dbwrite ». « hostname » et « port » font désormais partie des propriétés requises et peuvent être définies dans rfx.conf pour plus de commodité. Pour plus d'informations au sujet de la syntaxe et de l'utilisation, reportez-vous au Guide du programmeur RETL.
- Lors de la mise à niveau d'une base de données Oracle, la nouvelle base de données doit être testée à l'aide d'une session distincte de l'application RETL. En effet, seul un port à la fois peut être défini dans rfx.conf. L'autre solution consiste à utiliser l'application RETL avec l'option « -c » de manière à activer un autre fichier de configuration. Pour plus d'informations, reportez-vous au Guide du programmeur RETL.
- Les versions 11.x de l'application RETL peuvent nécessiter une quantité de mémoire physique plus importante pour certains flux que les versions 10.x.

Problèmes de compatibilité ascendante avec les versions 10.x

- Retek *déconseille* d'effectuer la mise à niveau vers RETL 11 à moins que le produit sur lequel RETL est exécutée dispose d'une certification pour RETL 11.

Known bugs in RETL 11.2.2

- Bug #511 – The port is not configurable for DB2 operators. This bug prevents RETL from running on a different server than the database.
- 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 #693 – TERAWRITE does not handle timestamp fields.
- Bug #744 – SORT does not work with newlines in data.
- Bug #759 – Invalid INPUT element results in StackOverflowError in flow parsing.
- Bug #781 – Duplicate OUTPUT causes warning and then NullPointerException.
- Bug #795 – Too few inputs to INNERJOIN throws an exception instead of showing an error.
- 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 #828 – DB2WRITE NullPointerException when used with a dataset.
- Bug #840 – CLIPROWS displays incorrect message about possible incorrect results when the input data is sorted in descending order.
- Bug#856 – If a record is altered by a BINOP operator after the record has been exported and then re-exported, the second EXPORT has incorrect fields because the record's cache is not cleared by the binop.
- Bug #917 – TERAFOREAD does not support multiple queries/partitioning.
- Bug #940 – An exception occurs when partitioning DB2WRITE with method=direct.
- Bug #941 – Parallel direct datafile ignored when partitioning.
- Bug #959 – Loops in INPUTs and OUTPUTs cause infinite loop in schema resolution.
- Bug #1069 – REMOVEDUP should warn if dataset is not sorted.
- 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.
- Bug #1209 – IllegalArgumentException exception in BINOP when destination is incorrect.
- Bug #1210 – A NullPointerException is thrown in BINOP when the destination field already exists but is NULL.
- Bug #1223 – RQST00000066282: FILTER fails when using the AND operation and IS_NOT_NULL on the left hand side of the expression.

Changes

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: xxx.xxxxxxxx

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, TERAFOREAD, 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-existing 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 TERAFOREAD

In RETL 11.2 and 11.2.1, TERAFOREAD 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/write.
- 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 sqldr.
- 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 len of the record equals the sum of the len 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 schemafilename 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 – Convolved 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 – leftouterjoin 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 – Dtty 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 moethod 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 algorithm 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 – Changelogcapture 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.