Documentation Addendum:

New Features in SQDR Plus 3.6x

April 2011

StarQuest Ventures, Inc.
PO Box 1076
Point Reyes Station, CA 94956
Telephone: 415-669-9619
FAX: 415-669-9639
Sales & Support: contact@starquest.com
URL: http://www.starquest.com/

What's New in this Release

This release of the StarQuest Data Replicator Plus introduces the following new features and improvements:

- New Installer launch application when running the SQDR Plus for iSeries installer on a Windows workstation. See the SQDR Plus Quick Start Guides for details.
- New text-based installer for SQDR Plus for UDB on UNIX systems.
- Ability to move or copy an incremental subscription between replication groups (requires SQDR 3.60 or later). See the SQDR Documentation addendum for details.
- Support for 4-Byte LSN (required for UDB 9.7. See Release notes for additional upgrade considerations)
- New Capture Agent startup options to support "fail-over"/"fall-back" scenarios
- Ability to take Subscription Offline/Online
- New "quiesce" shutdown option
- Enhancements to auto-journaling support (iSeries only)
- Option for CAMAINT to save support logs locally
- Changes to the Publish function
- Option to use *AFTER-Image (RRN-based) replication (iSeries only)
- Automatically send problem reports to StarQuest Support
- Ability to bypass staging transactions associated with a particular user
- Sequenced transaction support
- Support for partitioned tables (UDB only)
- Automatic deletion of inactive subscriptions
- Use "Block Insert" when updating staging tables (iSeries only)
- Provide warnings & suspend log readers if backlog detected

This document also contains the following Documentation Errata:

- Setting up User IDs (iSeries)
- Automatically starting the SQDR subsystem at IPL (iSeries)
- Manually Creating Journals for use with SQDR Plus (iSeries only)
- Using the Capture Agent Maintenance Utility – required authorities (iSeries)

New Features:

Text-based installer for SQDR Plus for UDB on UNIX systems

When installing SQDR Plus for UDB on UNIX systems, you can use either the GUI installer or the new text-based installer. The GUI installer is available if you are unfamiliar with the UNIX command line environment, but it requires access to an X-Windows environment. The text-based installer involves editing a configuration file with a text editor such as vi or gedit, and requires fewer system resources. In addition, support for some platforms (e.g. Linux for System z) is available only in the text-based installer, and future installer development will be oriented to enhancing the text-based installer.

You can use the text-based installer to upgrade from an existing installation performed with the GUI installer, after which future upgrades should also be done with the text-based installer.

Refer to the Quick Start Guide Using SQDR Plus for DB2 UDB on Linux/UNIX for step-by-step instructions for installing SQDR Plus on UNIX with the text-based installer.

Before you install:

The text-based installer expects that the database is already configured for the method of logging you plan to use. For instance, if you select managedLogs=true (the default), the database should already be configured for LOGARCHMETH1=USEREXIT. If the database is configured for circular logging (LOGARCHMETH1=NONE) or if you select managedLogs=true and the database is configured for anything other than USEREXIT, an error message is displayed and the installation is aborted.

If you are using sh/ksh/bash, modify /home/<*db2inst_owner*>/sqllib/userprofile; see Using SQDR Plus for DB2 UDB on Linux/UNIX.

Additional pre-requisites and pre-installation tasks are described in Using SQDR Plus for DB2 UDB on Linux/UNIX.

Editing the configuration file:

If this is a new install, copy setup.conf.template to setup.conf and edit it. Refer to the comments in setup.conf and to Using SQDR Plus for DB2 UDB on Linux/UNIX for detailed information.

If you are upgrading from an existing installation performed with the text-based installation, copy setup.conf.template to setup.conf and confirm that the value of INSTALLDIR (default /opt/StarQuest/sqdrplus) in setup.conf is correct. All other values will be obtained from the configuration file used during the original installation.

If you are upgrading from an installation performed with the GUI installer, copy setup.conf.template to setup.conf and edit it using the values used during the previous installation. Refer to the comments in setup.conf for detailed information.

Running the installation:

1. Log on to the UNIX/Linux computer as a user with root authority.

2. Set up access to DB2 by sourcing db2profile (sh/ksh) or db2cshrc (csh) in /home/<*db2inst_owner*>/sqllib.

3. Change to the directory where you extracted the installer and edited setup.conf.

4. Run the installation:

   # ./setup [-v] [-c]

   -v = Verbose (optional)
   -c = Recreate control tables during an update (optional; ignored during a new installation)

The installer will attempt to verify whether this is a new installation, an update from a previous text-based installation, or an update from a previous GUI installation. It will ask for confirmation of the type of installation before proceeding.

You will be prompted for the password of the DB2 user configured in setup.conf; this user and password are used during installation for creating control tables and registering stored procedures, and are used during operation by SQDR Plus. The same DB2 user will be used during uninstallation, at which time you will be prompted for its password.

After installation, examine the logs in /var/sqdrplus/<DATABASE>/logs for unusual errors. Some errors are expected; for instance, after a new installation, you can ignore errors in AgentUtilR.log.

Changing the DB2 user password:

If the password of the DB2 user should change at any time after installation of SQDR Plus, run the following command to notify SQDR Plus of the change:

1. cd /opt/StarQuest/sqdrplus (or wherever you installed to)

2. ./change_password

You will be prompted for the new password.

Uninstalling SQDR Plus:

1. Log on to the UNIX/Linux computer as a user with root authority.

2. Set up access to DB2 by sourcing db2profile (sh/ksh) or db2cshrc (csh) in /home/<*db2inst_owner*>/sqllib.

3. cd /opt/StarQuest/sqdrplus (or wherever you installed to)

4. Run the uninstallation:

   ./uninstall [-v]-v – verbose (optional)

You will be prompted for the password of the DB2 user that was configured in setup.conf.

You will be prompted whether to remove the SQDR Plus control tables.

After uninstalling the product, the bin, lib, conf, and logs subdirectories of /var/sqdrplus/<DATABASE> are renamed with a prev suffix – e.g. bin is renamed to bin.prev to bin.prev, etc.

You can examine the results of some of the actions of uninstalling by examining the logs in /var/sqdrplus/<DATABASE>/logs.prev. In addition, conf.prev/sqagent.properties and logs.prev/install_history.log may be useful as a reference of your previous configuration if you choose to reinstall SQDR Plus.

New Capture Agent startup options to support "Fail-over"/"Fall-back"

CAMAINT now includes a Recovery menu which allows the SQDR Plus Administrator to change the startup mode for the Capture Agent and take an individual incremental replication subscriptions offline or online.

The Recovery menu is shown below.

Agent Recovery Menu

1. Set Startup Mode (Current: "Normal")
2. Take Subscription Online
3. Take Subscription Offline
4. Quiesce Capture Agent

The Startup Mode affects the behavior of the Capture Agent, the next time it is started. The Online/Offline functions take effect immediately. The following sections detail each feature in detail.

Set Startup Mode

The default Startup Mode for the Capture Agent is Normal. Before changing the Startup Mode, the Capture Agent must be stopped.

When to change the Startup Mode

The Startup Modes are useful in situations where the source database is moved from a production system to a backup system ("fail-over") or where SQDR operations are restored on a primary server ("fall-back"). Both operations can be performed without the need for SQDR to run baseline replications. This option is termed a “Warm” startup mode. Another new startup option, “Cold” startup, will force all subscriptions to be flagged as requiring new baselines. This “Cold Start” option is useful when the host control tables are intact and in agreement with the SQDR client(s), but the User source and destination database tables are no longer be assumed to be synchronized. For example, this situation might occur in the event of unrecoverable log damage, necessitating a restore. Once the Capture Agent has completed the startup-time recovery, it resets the startup mode for the next invocation to “Normal”.

The Set Startup Mode has three options:

1. Startup with fully recovered control and staging tables.
2. Force all subscriptions offline ("online" needed after restart).
3. Normal Startup.

Normal Startup

SQDR Plus assumes that the state of its log, staging tables and log positions are still current and accurate. However, at startup, the Capture Agent checks for source table structure changes. The Startup Mode should be set to Normal except when a "fail-over" or "fall-back" operation is being performed.

“Warm” Startup with fully synchronized control and staging tables

SQDR Plus assumes that logging tables and staging tables are in a working state and that it only needs to recover the current log position.

This option is useful when copying the entire SQDR Plus schema (e.g., library) to a new system, or other types of complete restore operations where the consistency of the database is guaranteed. SQDR Plus

skips any journal entries showing that tables having been deleted and re-created by moving its log/journal read position to "now".

As in a Normal startup, the Capture Agent validates source table structures. If auto-journaling is enabled and SQDR Plus discovers that a source table is no longer journaled, it will automatically try to restore journaling. Also, grants to the views on the staging tables are made to match the source tables.

“Cold” Startup

SQDR Plus makes no assumption about the consistency of log and staging tables. It skips any log/journal changes and takes all subscriptions "offline". The user can then manually move subscriptions "online" in a controlled fashion once SQDR Plus is restarted. Taking a subscription online will cause a new baseline replication to be required. (See the following section re. Offline/Online processing for more details.)

As with the *Startup with fully recovered control and staging tables* mode, metadata verification occurs, journaling is restored, if necessary, and grants to the views on the staging tables are made to match the source tables.

In the latter two cases, SQDR Plus assumes that the source and destination tables are in sync at the time of the "fail-over" or "fall-back" operation, and that only its log position is to be considered invalidated.

Ability to Take Subscription Offline/Online

The ability to take a subscription offline/online should be used when significant changes are made on the source tables. In this case, moving subscriptions to offline will suspend staging of changes and will require a new baseline in order to make new changes available. This option is particular useful if a set of tables is restored or loaded with a large number of new rows.

This functionality allows significant changes to be made to a source file or a set of source files (e.g., dropping/re-creating source tables or adding/removing members from tables) without requiring the user to re-create the subscriptions in SQDR.

Taking a subscription offline will flag a given source table as being unavailable for normal Capture Agent processing (e.g., staging changes), until further notice. A subscription in the offline state appears to the SQDR client as if there is no activity on the subscribed table. This allows a database administrator to restore the table to a desired version without the Capture Agent processing and staging all records, which may cause significant delay on large tables.

Once the restore operation is complete, the administrator uses the Take Subscription Online option to notify the Capture Agent to resume processing for the subscribed table. At this point the Capture Agent will compare its saved metadata for the table to its current state. If the structure of the subscribed table is unchanged, the SQDR client is notified that only a new baseline snapshot is required. If there have been structural changes or if the subscribed table no longer exists, a "table altered or deleted" notification is generated and the subscription must be deleted and re-created. If the subscription is already online when you issue the online request, no changes occur.

(iSeries only) If the table is no longer journaled after the restore, SQDR Plus will attempt to restore journaling for the table. If the attempt to start journaling fails, the subscribed table will remain "offline".

The Online and Offline operations can be performed on multiple libraries (or schemas) and tables by using iSeries CL wildcards (when using the VRYSUB command) or SQL search patterns (when using CAMAINT or invoking Java directly)

The Online/Offline commands are available in the following ways:

- Use the Agent Recovery Menu of CAMAINT

  Agent Recovery Menu

  1. Set Startup Mode (Current: "Normal")
  2. Take Subscription Online
  3. Take Subscription Offline

  Enter selection or press enter to continue

- VRYSUB command (iSeries only)

  From an iSeries command line or, within a CLP, add the SQDR Plus schema to the library list and run the VRYSUB command. Provide the name of the library where the source file is located, the name of the source file, the desired status (*OFFLINE or *ONLINE) and the SQDR Plus schema (typically SQDR), as shown in the example below.

```
ADDLIBLE SQDR
VRYSUB <F4>

                    Vary Subscription (VRYSUB)

Type choices, press Enter.

Library  . . . . . .   PRODLIB      Name, generic*, *ALL
File . . . . . .       CUSTLIST     Name, generic*, *ALL
Status . . . . . .     *OFFLINE     *OFFLINE, *ONLINE
Install Library  . . . . . .   SQDR         Character value
```

- Start AgentUtil with the –online or –offline argument

```
java –jar AgentUtil.jar [-online | -offline ] <agentSchema>
      <schemaPattern> <namePattern>
```

Examples:

Qshell (iSeries):

```
$ cd /home/sqdr
$ java -jar AgentUtil.jar -online SQDR MYLIB FILE%
```

SQDR Plus/UDB (Windows):

```
C> cd \Program Files\StarQuest\SQDRPLUS
C> java -jar "%CLASSPATH%:AgentUtil.jar" -online SQDR MYSCHEMA ^
FILE%
```

SQDR Plus/UDB (UNIX):

```
$ cd /opt/StarQuest/sqdrplus
$ java -jar $CLASSPATH:AgentUtil.jar -online SQDR MYSCHEMA FILE%
```

New "quiesce" shutdown option

A new option "Quiesce Capture Agent" has been added to the Recovery Menu of CAMAINT.

When this option is selected, CAMAINT makes a Quiesce call to the Capture Agent. Upon receipt of the Quiesce request, the Capture Agent terminates the pruning, publishing, and client notification threads. It then notifies each replication worker (one per journal/log being processed) to quiesce its operation. The replication worker will process any outstanding log/journal records until a call to the log/journaling API returns no records. The replication worker will then enter a quiesced state, where it will remain until the capture agent is stopped.

To monitor the state of the Capture Agent's replication workers, choose option 3 ("Display current status") from the CAMAINT menu. Once all replication workers have entered quiesced state, you can be assured that all log/journal entries have been processed and placed in the capture agent's staging tables, assuming that no new activity has been allowed on the subscribed tables after the replication worker entered quiesced state.

If a replication worker experiences a resource failure while processing in this mode, the Capture Agent will restart the worker and, barring other failures, will process any remaining log/journal entries, eventually entering quiesced state.

Enhancements to auto-journaling support (iSeries only)

Previously, auto-journaling was limited to journaling physical files in libraries that were co-located in the same ASP (Auxiliary Storage Pool) as the SQDR Plus Capture Agent schema. SQDR Plus now supports auto-journaling for multiple ASPs.

When a non-journaled physical file is selected for incremental replication, the SQDR Plus Capture Agent will create a new library in the same ASP as the physical file. This library will be named <agentSchema><NN>, where agentSchema is the name of the Capture Agent schema and userid (default SQDR) and NN is the ASP number. For example, if the Capture Agent schema is SQDR and the physical file is located in ASP 3, a library named SQDR03 will be created in ASP 3.

The library and its contents will be owned by the Capture Agent userid, and will contain the journal created by the Capture Agent for auto-journaling.

Journal receivers for the journal will be created in the Journal Receiver library specified during installation. If you plan to use auto-journaling, we recommend considering ASP layout before installing SQDR Plus. For best performance, the Journal Receiver library should be located in an ASP other than the ASP containing the files or tables to be replicated.

Option for CAMAINT to save support logs locally

Due to firewall policies or other restrictions, the database server running SQDR Plus may be unable to use ftp to send SQDR Plus support files directly to StarQuest.

The new Support menu is as follows:

Support Menu

1. Send SQDRPlus Logs (ftp)
2. Change ftp server (ftp.starquest.com)
3. Change ftp userid (sqdrdrop)
4. Change ftp password
5. Save SQDRPlus Logs to local file system (as zip file)
6. Set directory for saving logs locally (\temp)
7. Specify id of user that installed SQDRPlus (QSECOFR)

Make a selection or hit RETURN to continue.

Use option 5 to save SQDR Plus logs to the local file system so they may be forwarded using another method. This option will save the files to the iSeries file system (when CAMAINT is run on the iSeries server), the DB2 UDB server's file system (when running SQDR Plus/UDB's CAMAINT), or the local workstation's file system (when running SQDR Remote Admin's CAMAINT).

You may also specify the target directory for the zip files (option 6).

Changes to the Publish function

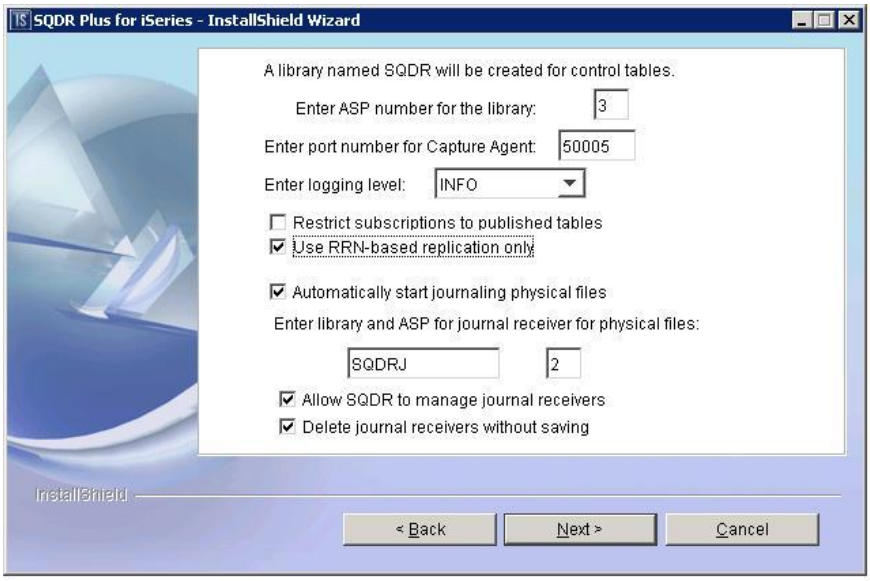
The Publish function no longer requires that an exclusive lock be acquired on a table before being considered published. If an attempt to lock the table for shared update fails, the subscription will be published with a "*" character next to the state of "Published" to indicate that a lock was not obtained prior to starting to stage changed data.

If the original more limiting behavior is desired, it can be configured by setting the keyword retryPublishLock=true in the sqagent.properties file; in this case the publication will be displayed as "Subscribed", and the lock operation will be retried every 10 minutes. Once successfully locked (and immediately released), the state will be updated to "Published".

Option to use *AFTER-Image (RRN- based) replication (iSeries)

This option is useful for subscribing to tables whose journaling is configured with Record Images=*AFTER. For best results, we recommend that tables be journaled for Record Images=*BOTH (the system writes both before and after images to the journal for changes to records in the table), but this option is available if an existing journaling configuration cannot be changed.

You may choose this option by setting alwaysUseRowId=true in sqagent.properties, or by selecting the “Use RRN-based replication only” checkbox during installation or custom update:



Limitations: You cannot specify "criteria" when configuring incremental subscriptions for tables whose journaling is configured with Record Images=*AFTER.

Ability to bypass staging transactions associated with a particular user (SQDR Plus/UDB 3.64, SQDR Plus/iSeries 3.69 and later)

Configuring the filterUserId property in sqagent.properties specifies that the Capture Agent should bypass staging transactions associated with the specified userID.

The property can have a value of a single userID. With the property in place, the Capture Agent examine each commit record; if the user-id in the commit matches the "filterUserId", the commit is handled like a rollback, and any staged data is removed.

Restart the Capture Agent after modifying sqagent.properties.

Example:
filterUserId=myuser

Sequenced transaction support (SQDR Plus 3.65 and later)

.

Setting the useTxSequence property in sqagent.properties to a value of true specifies that the Capture Agent should maintain the order of transactions. The default value of this property is false, which means that replications are consistent at transaction boundaries.

Restart the Capture Agent after modifying sqagent.properties.

This property is useful if the target table has any foreign constraints defined on it, such as the target of an incremental replication defined with "unique constraints." The setting preserves the order of the source updates on the target side and is useful if you are using triggers on the target and need updates to arrive in the order they were performed at the source. The update order is always maintained within each table; this setting is useful if the update order between tables is important.

An example is a single transaction that updates table A, then table B, then table A, then table B. Without this setting, the changes would be applied as A,A,B,B, Commit; with this setting, the order would be A,B,A,B, Commit.

Example:
useTxSequence =true

Support for partitioned tables (UDB only; SQDR Plus 3.65 and later)

SQDR Plus for UDB supports incremental replications for partitioned tables in DB2/UDB LUW 9.1 and later.

Adding empty partitions and subsequently populating them requires no special considerations. SQDR Plus will detect newly-added partitions, and automatically starts processing transaction log entries for the new partition.

Considerations for ATTACH/DETACH partition:

SQDR Plus will detect when partitions are DETACHED, and when partitions are ATTACHED (and INTEGRITY is set for the table). These events will cause SQDR Plus to notify the SQDR client that a new baseline is required. If the SQDR client (v3.62 and later) is configured to “ignore deletes”, the DETACH notification is ignored. There may be other considerations associated with “ignore deletes” that should be considered when using this option.

Automatic deletion of inactive subscriptions

Previously, SQDR Plus detected inactive subscriptions and sent a notification email (type SEVERE) after seven days of inactivity. The system administrator could then take some action to resolve the issue, either deleting the subscription and subscriber with CAMAINT, or investigating why the client computer was not communicating with the iSeries server. However, if no action was taken to resolve the issue, the backlog of staged data could continue to grow.

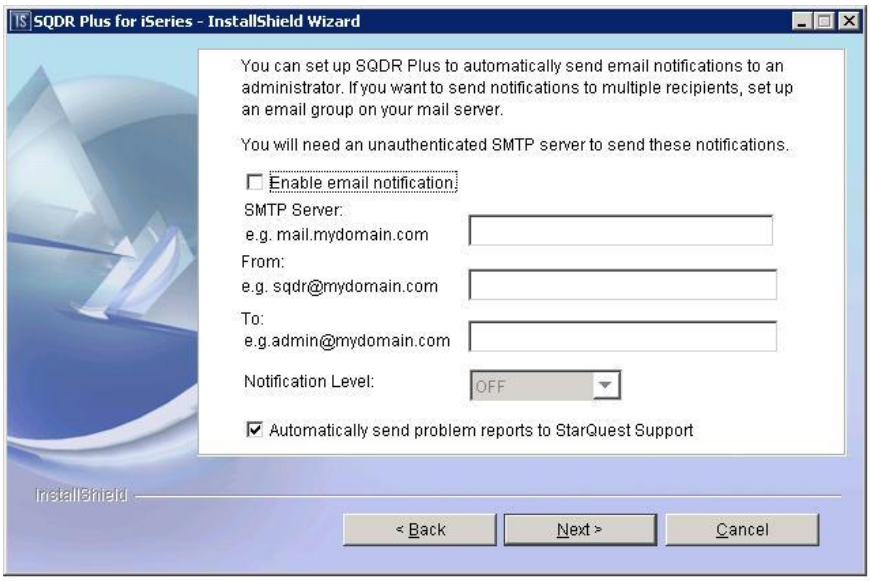
SQDR Plus will now send the notification email after seven days, and then automatically delete the subscription and subscriber after a grace period of one additional day expires, allowing purging of staged data.

You can configure the grace period in sqagent.properties with a new property "clientDeleteGrace". Default value is 1 (day). A value of 0 will configure SQDR Plus to behave as it did previously: send the seven day notification, but do not automatically delete subscriptions and subscribers.

You can recover deleted subscriptions by using the "Reset I/R Group" function from the SQDR Data Replication Manager; a new baseline will be performed.

Automatically send problem reports to StarQuest Support

The Capture Agent can now automatically send problem reports via email to StarQuest Support. This option is enabled by default. To disable it, set starquestNotification=OFF in sqagent.properties or uncheck the checkbox during installation or custom update:



Use "Block Insert" when updating staging tables (iSeries only)

The SQDR Capture Agent now uses "block insert" to improve staging performance. This feature results in significant performance improvements when many rows are updated/inserted/deleted in a single table or a small number of tables. However, it may result in poorer performance if a single row is touched in many hundreds of tables; in this case, you may choose to disable the feature by setting batch=false in sqagent.properties.

There are two new properties in sqagent.properties:

Batch=true (default) or false (previous behavior)
batchSize= the number of rows to allow in a single "insert" batch (default 100)

Provide warnings & suspend log readers if backlog detected

SQDR now provides a mechanism for monitoring backlogs and will send a warning notice and suspend log readers if necessary to avoid excessive usage of the table space where the staging tables reside. The default mechanism uses a simple row-count based throttling mechanism, but is user-extensible.

By default, the Storage Monitor uses a simple SQL statement to count the number of rows in complete transactions available in the staging tables. You can define alternate criteria with the smSQL property in sqagent.properties; this should be specified as an SQL statement which returns an integer number.

When the number of rows exceed a certain value (defined as smWarningLevel), then a warning notification is issued.

When the number of rows exceeds the smThrottleLevel value, another warning notification is issued and all log readers (replication workers) are paused. The log readers resume when the number of rows drop below the smThrottleLevel value. When the "storage monitor" has throttled staging, pruning runs at a higher frequency than usual (using the "storage monitor" interval).

The functionality is controlled by the following sqagent.properies settings:

SmSQL - a SQL statement used to monitor staging activity; it should return a single integer or bigint value. Note that the default SQL statement (used when smSQL is not specified) is valid only when SQDR Plus is monitoring only one journal.

Examples (iSeries):

```
smSQL=SELECT SUM(ROWCOUNT) FROM SQDR.SQ_TRANSACTIONS WHERE ROWCOUNT > 0 AND
TXID IS NOT NULL AND TXID <= (SELECT MIN(MAXTX) FROM SQDR.SQ_READERS)

smSQL=select sum(number_row_pages) from qsys2.systablestat where
table_schema='SQDR'
```

smInterval (default 0 (off)) – Frequency in minutes of when to run the storage monitor. 0 means "off".

smWarningLevel (default 450000) - Any integer value above this value will cause a notification/log message.

smThrottleLevel (default 500000) - Any integer value above this value will pause all log readers (replication workers) and cause a notification/log message.

Note that in some cases this feature may have undesirable side effects. For example, if staging is suspended because of the "storage monitor" condition, the storage used by the journal receivers (iSeries) or archived transaction logs (UDB) will grow instead; this may be less desirable, especially if SQDR Plus is staging only a small portion of the data in the receivers/logs. You should analyze your system and compare the amount of available storage for the library or container holding the SQDR storage tables with the amount of available store for the journal receivers or logs to determine whether or not this feature is suitable for your environment.

Documentation Errata:

Setting up User Ids (iSeries) (p15)

Capture Agent User

The Capture Agent is an “always-on” task that is started when the subsystem into which it is installed is started. The Capture Agent user account is created during the installation if the specified user account does not already exist, and is deleted when SQDR Plus is uninstalled. Therefore, you may want to create a new user account that is dedicated to the Capture Agent user. During installation the Capture Agent user ID also is granted permission to modify the appropriate tables in the control table schema and to access the Journal Reader service program.

The Capture Agent user ID can be an unprivileged user (*USER) but it does require logon permission (PASSWORD should **not** be *NONE). We recommend setting its Initial Menu (INLMNU) to *SIGNOFF to prevent the use of this ID for interactive signon.

The default action of the installer is to create a new user SQDR with its Initial Menu (INLMNU) set to *SIGNOFF. The dialog to specify the Capture Agent User contains a field to enter a password. Leaving this field blank will create a user with password *USRPRF (i.e. the password matches the user name), but the Initial Menu setting will prevent interactive signon. If you enter a password, the installer will send a CHGUSRPRF command to the iSeries server; be sure to review the resulting message that is displayed, as the installer will continue whether or not the iSeries machine accepted the password change.

During a custom update, you may enter the password in this dialog to change the password for the Capture Agent user ID. As with the initial installation, the installer will continue whether or not the iSeries machine accepted the password change, so be sure to review the resulting message.

Journal Reader Service Program Owner

The Capture Agent runs as an unprivileged user, but it needs access to the journal receivers of all tables that are to be replicated. This is accomplished by invoking the JRLREADER service program. This service program is created as User profile (*OWNER), which indicates that it runs under both the user profile of the current user and the user profile of the owner.

The owner ID for the Journal Reader Service Program should be a privileged user; its PASSWORD can be set to *NONE to prevent the use of this ID for signon. It should have *ALLOBJ and *SECADM authority, or at least have read authority to all tables and journal receivers that will be replicated, regardless of who the owner of the table is and what user ID is used for the ODBC connection from the SQDR client.

The default action of the installer is to create a new user SQDRADMIN with PASSWORD set to *NONE.

Automatically starting the SQDR subsystem at IPL (iSeries) (p33)

Note that TCP/IP services should be started before the SQDR subsystem.

Using the Capture Agent Maintenance Utility (iSeries) – required authorities (p35)

To run CAMAINT as a user other than the SQDR user or a user with SECOFR authority, the following grants must be made. Note that these grants will need to be repeated after an update of SQDR+/iSeries.

Note that *PUBLIC already has *USE on SQDR *LIB

1) Use WRKOBJ (option 2), or GRTOBJAUT to grant *USE for the following objects:

SQDR/CAMAINT *PGM
SQDR/CAMAINT *CMD
SQDR/CAPATH *DTAARA

GRTOBJAUT OBJ(SQDR/CAMAINT) OBJTYPE(*PGM) USER(*MYUSER*) AUT(*USE)
GRTOBJAUT OBJ(SQDR/CAMAINT) OBJTYPE(*CMD) USER(*MYUSER*) AUT(*USE)
GRTOBJAUT OBJ(SQDR/CAPATH) OBJTYPE(*DTAARA) USER(*MYUSER*) AUT(*CHANGE)

2) Use WRKAUT or QSH/chmod to grant *RX (read and execute) for the following IFS objects:

/home/sqdr
/home/sqdr/AgentUtil.jar

WRKAUT OBJ('/home/sqdr')
WRKAUT OBJ('/home/sqdr/AgentUtil.jar')
Choose 1 (Add user), enter the user name, and *RX for Data Authority

Creating Unique Journals (iSeries) (p45)

Instead of using SQDR-created journals and receivers, you can create your own unique journals, journal receivers, and journal receiver libraries. This offers more control in configuring access security and location of the objects.

Note that a single journal can be used for multiple physical files. You can use any valid name for the journal and receiver; however, avoid the names QSQJRN and QSQJRN*nnnn*, as these are the default names used when the OS/400 creates journal and journal receivers via the CREATE SCHEMA or CREATE COLLECTION SQL statements.

The following example illustrates how to create a journal called "MYLIB/MYJRN" and journal receiver "MYLIB1/MYRCV0001" and how to associate the physical file "MYLIB/MYFILE" with the journal. Often the journal is placed in the same library as the file to be journaled, but if not, both the journal and the file must reside in the same ASP. This requirement does not extend to the journal receiver library; it may be useful to place the journal receiver library in a separate ASP for performance and recovery purposes.

CRTJRNRCV JRNRCV(*MYLIB1*/MYRCV0001) TEXT('receiver for physical files')

CRTJRN JRN(*MYLIB*/MYJRN) JRNRCV(*MYLIB1*/MYRCV0001) RCVSIZOPT(*MAXOPT2 *RMVINTENT) TEXT('journal for physical files')

STRJRNPF FILE(*MYLIB*/*MYFILE*) JRN(MYLIB/MYJRN) IMAGES(*BOTH)

To stop journaling, issue the **ENDJRNPF** command. Once all journaling on the physical files has ended, the previously associated Journal and Journal Receiver(s) may be deleted.

Revisions on procedure to remove SQDR Plus/iSeries (p47)

Note: When SQDR Plus is removed, the uninstall job may not be able to remove all objects created during the course of product usage.

If any user files are still configured as auto-journaled or published when the SQDR Plus product remove job is run, the SQDR library containing the SQDR Journal and any Receiver libraries associated with User ASP(s) may remain. The associated objects will be owned by QDFTUSR. In this case, manually use ENDJRNPF to stop journaling the user files, and then delete the unused SQDR Journal and Receiver objects.

To avoid this situation, before running the product remove job, insure that no files remain published or subscribed to, and that no subscribers still remain. Use CAMAINT to “unpublish” and to delete remaining subscribers and subscriptions.

Revisions on procedure to remove SQDR Plus/UDB (p57-58)

NOTE: If the SQDR Plus installer changed the logging type of the database from circular to Exit Program, change it to either archive or circular, or install another exit program, after removing SQDR Plus.

Removing the SQDR Plus/iSeries Exit Program (iSeries) (p45-46, “Maintaining Journals and Log Files”)

SQDR Plus installs an exit program to prevent the premature removal of journal receivers it needs when working with subscribed tables. If you wish to manage journal receivers yourself, use the following instructions to remove the exit program.

- Make sure that /home/sqdr/sqagent.properties is configured as follows:

  managedLogs=false
  autoJournal=false

- Call the CALOGMGT program to remove any current exit program registration known to the SQDR application.

```
CALL PGM(SQDR/CALOGMGT) PARM('SQDR' 'U')
```

- Use the WRKREGINF command to delete any remaining occurrence of the registration program for SQDR/CADLTRCV left from a previous installation/update:

1. Enter WRKREGINF at a command line.
2. Page down to a Journal Receiver exit point:

```
QIBM_QJO_DLT_JRNRCV DRCV0100
```

3. Enter 8 in the Opt field to the left of the exit point entry and press Enter.
4. At the Work with Exit Program display, identify the exit program entry for “CADLTRCV” in library “SQDR” and enter a 4 (Remove).
5. Press Enter.

Format of the SQ_STATISTICS control table (p48 under “Performance Suggestions”)

The SQDR Plus software maintains a table called SQ_STATISTICS which stores statistical data related to the operation of the software. Understanding how this data is collected and used can be helpful in evaluating the performance of the software on the host database.

The statistics in the SQ_STATISTICS table are periodic snapshots of in-memory statistics maintained by the SQDR Plus Capture Agent. These in-memory statistics are written/flushed to the SQ_STATISTICS table every two minutes.

The data contained in this table can be easily viewed in CAMAINT by choosing Option 4 Display current statistics or Option 5 Display all statistics. The statistics reflect the activity for the last 7 days, in hourly increments. Displaying the current statistics shows all of the statistics in the current hour. For example, if you check the statistics at 10:18 AM, the statistics shown are those that were captured in the last 18 minutes of activity (between 10:00AM and 10:18AM). If you would like to view statistics that were recorded before the current hour, use the Display all statistics option.

Active replications:
The TOTAL "Active replications" value is the number of source tables managed by a particular journal (and log reader). The journal name is indicated in parentheses. AVG and MAX values are also displayed.

Lag time:
The "Lag time" TOTAL represents in seconds the difference between the time when the source table received updates and the time when the data is moved to the staging table. If the lag time is close to zero, no significant backups have occurred during the hour. A large value indicates that the SQDR Plus log reader is lagging behind in processing changed data and may indicate that the source table received a very large number of changes in a short period of time and may be better suited for snapshot replication.

If "Lag time" MIN value is 2147483647and MAX value is –2147483648, then no statistics have been collected for the time period. If MIN/MAX have other values, those are the true MIN/MAX values collected for the time period covered by the statistic "bucket". An AVG value is also displayed.

Records processed:
The "Records processed" value is the number of records processed for the particular journal reader. This count includes records that were both committed and roll-backed and does not necessarily indicate the number of new updates that have occurred on the source table(s). If this value is lower than expected (e.g., the source tables for this journal received 100 new records in the last hourly unit of time, but the records processed value is only 10), it is reasonable to expect that this would be associated with a non-zero lag time.

Staged Rows:
The "Staged Rows" value is the aggregate number of rows left in the staging tables at the end of a pruning operation. The default behavior is for the Capture Agent to prune the staging tables every 30 minutes, so this value is typically updated twice every hour.

Prune Cycles:
The number of times pruning has run in the current hour. For default pruning settings (every 30 minutes), this value will be 0, 1, or 2.

Prune duration:
The shortest, longest and most recent pruning duration times (MIN, MAX and TOTAL values, in seconds) during the hour bucket.

Rows pruned:
The number of rows pruned in the current hour.

In addition to displaying statistics with CAMAINT, you can use an ODBC-enabled data analysis application and an ODBC driver such as StarSQL (or a JDBC application and a JDBC driver such as StarSQL for Java) to analyze the SQ_STATISTICS table.

The SQ_STATISTICS table contains the following fields:

COMPONENT: This is the qualified journal name or 'Pruner'. VARCHAR
STATNAME: Identifies the type of statistic (Lag time, Records processed, Active replications, Staged Rows, Prune Cycles, Prune duration, Rows pruned). Note that these names are case-sensitive. VARCHAR
STATDATE: The date of the statistic entry. DATE
STATHOUR: The hour of the statistic entry (0-23) INTEGER
STATTYPE: The type of the statistic (e.g., MIN, MAX, AVG, TOTAL). VARCHAR
INTVALUE: The value of the statistic (except for AVG values). INTEGER
DOUBLEVALUE: The value of the average (AVG) statistic. DOUBLE

New Appendix A:
OS/400 Authorities, Commands and API usage by SQDR Plus (iSeries)

The user “SQDR” uses the following Authorities and Locks:

Journal Authority

*OBJOPR and some data authority other than *EXECUTE

Journal Library Authority

*EXECUTE

Journal Lock

*SHRRD

The Journal Reader program (SQDR/JRNLREADER *SRVPGM) is owned by the user SQDRADMIN and uses *OWNER authority in order to execute the following commands:

```
QDBRTVFD
QjoRetrieveJournalInformation
QjoRtvJrnReceiverInformation
QjoRetrieveJournalEntries
QjoDeletePointerHandle
QJOSJRNE
QMHSNDM
QMHRTVM
ENDJRNPF
GRTOBJAUT OBJTYPE(*LIB) AUT(*CHANGE) … to SQDR user
GRTOBJAUT OBJTYPE(*FILE) AUT(*USE) ) … to SQDR user
GRTOBJAUT OBJ(" + changeView.mSystemSchemaName + "/" +
changeView.mSystemName + ") OBJTYPE(*FILE) REFOBJ(" +
table.mSystemSchemaName + "/" + table.mSystemName + ")
REFOBJTYPE(*OBJTYPE)"
```

New Appendix B:
sqagent.properties sample (iSeries)

The following is the contents of the template sqagent.properties file from which the installer creates sqagent.properties during the initial installation. Refer to /home/sqdr/sqagent.template to view the latest version of the template.

Items that are not commented out are properties that are configured by the installer based on user input during the installation; items that are commented (leading #) are advanced properties that can be configured by the user; the default value is displayed here.

```
# SQDR Plus for iSeries
# Capture Agent Config Properties
#
# CAPTURE AGENT PARAMETERS
#
controlDbSchema=SQDR
logLevel=INFO
port=50005
requirePublication=false
alwaysUseRowId=false
receiverLibrary=SQDR
# pollSleepTime=5000
# pruneInterval=30
# grantees=
# numControlDbConnectionsPerVm=10
# singleTrans=true
# useTxSequence=false
# clientDeleteGrace=1
# filterUserId=myuser
#
# batch = use Block insert when updating staging tables
# batchsize = the number of rows in a single block insert
# batch=true
# batchSize=100
#
# Space Monitor Parameters
# smSQL=the sql used to monitor staging activity
# returns a single integer or bigint value that reflects usage.
# sminterval=frequency of monitor in minutes.  0 means "off".
# smInterval=0
# smwarning=issue a notification/log message if value returned by smSQL
exceeds this value
# smWarningLevel=450000
# smThrottleLevel pause all log readers (replication workers)
#  and issue a notification/log message if value returned by smSQL exceeds
this value
# smThrottleLevel=500000
#
#
# OPERATIONAL NOTIFICATIONS PARAMETERS
#
notificationLevel=OFF
smtpServer=
to=admin@mydomain.com
from=sqdrplus@mydomain.com
```

```
starquestNotification=SEVERE
#
#
# CHANGE DATA LOGGING PARAMETERS
#
managedLogs=false
autoJournal=false
deleteLogsWithoutSave=true
# maxLogFileSize=512000
# maxNumLogFiles=5
#
#
# CHANGE DATA NOTIFICATION PARAMETERS
#
# broadcastAddress=192.1.1.255
# udpPort=2728
# minimumNotificationDelay=5000
#
#
```

New Appendix B:
sqagent.properties sample (UDB)

The following is the contents of the template sqagent.properties file, from which the installer creates sqagent.properties during the initial installation. Refer to /opt/StarQuest/sqdrplus/DBCONFDIR/conf/sqagent.properties to view the latest version of the template.

Items that are not commented out are properties that are configured by the installer based on user input (or, in the case of the text-based installer, by setup.conf) during the installation; items that are commented (leading #) are advanced properties that can be configured by the user; the default value is displayed here.

```
# SQDR Plus for DB2 UDB
# Capture Agent Config Properties
#
#
# CAPTURE AGENT PARAMETERS
#
controlDbSchema=SQDR
logLevel=INFO
port=50005
requirePublication=false
# pollSleepTime=5000
# pruneInterval=30
# grantees=
# numControlDbConnectionsPerVm=10
# singleTrans=true
# useTxSequence=false
# clientDeleteGrace=1
#
# Space Monitor Parameters
# smSQL=the sql used to monitor staging activity
# returns a single integer or bigint value that reflects usage.
# sminterval=frequency of monitor in minutes.  0 means "off".
# smInterval=0
# smwarning=issue a notification/log message if value returned by smSQL
exceeds this value
# smWarningLevel=450000
# smThrottleLevel pause all log readers (replication workers)
#  and issue a notification/log message if value returned by smSQL exceeds
this value
# smThrottleLevel=500000
#
#
# DB2 UDB CONNECTIVITY PARAMETERS
#
sourceType=UDB
userId=
password=
controlDbDriver=com.ibm.db2.jcc.DB2Driver
controlDbUrl=jdbc:db2:127.0.0.1:446/RDBNAME:driverType=4;deferPrepares=false;
useAuthentication=true
TABLEINFO=false
#
#
# OPERATIONAL NOTIFICATIONS PARAMETERS
```

```
#
notificationLevel=OFF
smtpServer=
to=admin@mydomain.com
from=sqdrplus@mydomain.com
starquestNotification=SEVERE
#
#
# CHANGE DATA LOGGING PARAMETERS
#
managedLogs=false
deleteLogsWithoutSave=true
archivedLogPath=
# maxLogFileSize=512000
# maxNumLogFiles=5
#
#
# CHANGE DATA NOTIFICATION PARAMETERS
#
# broadcastAddress=192.1.1.255
# udpPort=2728
# minimumNotificationDelay=5000
#
#
```