

Information and/or specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein are covered by one or more U.S. or foreign patents. This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc. Complete Confidentiality, Trademark, Copyright and Patent notifications can be found at: <http://www.tridium.com/galleries/SignUp/Confidentiality.pdf>. © 2013 Tridium, Inc.

JACE, Niagara Framework, Niagara AX Framework and the Sedona Framework are trademarks of Tridium, Inc.

JACE Data Recovery Service (SRAM support)

Starting in AX-3.6, NiagaraAX support was added for “battery-less” JACE operation, where a JACE uses capacitor-charged SRAM (static random access memory) to preserve RAM-resident data when a power outage occurs. This includes station data not yet been committed to non-volatile flash memory.

- Initially, this applied only to a QNX-based JACE controller with an installed *SRAM option card*. Previously, such controllers were always considered “battery-less”, where any installed backup battery was removed when the option card was installed. However, starting in an AX-3.6 maintenance release, a JACE controller can utilize both SRAM and a backup battery.
- Starting in early 2012, new QNX-based JACE models have “onboard” SRAM as standard—no option card required. This includes the newest JACE-3E series controller (introduced in 2013), plus all the NPM6E processor-based series (JACE-6E and “retrofit board” JACE-603 and JACE-645 controllers). Both the JACE-6E and JACE-3E controllers ship “battery-less”—however, you can *optionally* install a NiMH backup battery (identical to the one in JACE-2/6 series controllers).
- AX-3.7 and later supports all the new controller models (for JACE-3E support, build 3.7.105 or later needed), and also all controllers with an installed SRAM option card.

Note: *The Security JACE platform does not include SRAM, nor does it support the SRAM option card / DataRecoveryService in any build of NiagaraAX.*

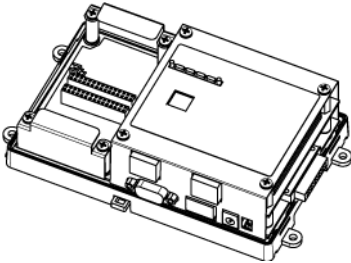
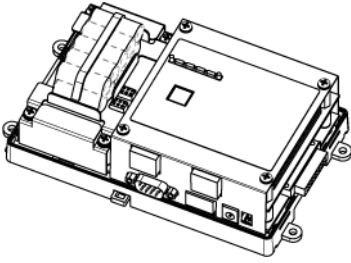
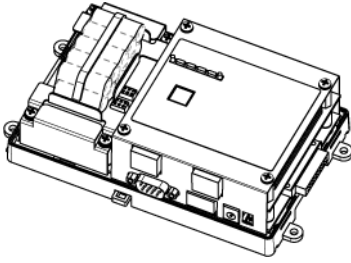
For hardware mounting details, refer the installation document that ships with each SRAM option card or SRAM-equipped controller. This document summarizes usage scenarios and software operation details of the SRAM memory feature.

The following sections provide more details:

- “SRAM and/or battery options” on page 2
- “Battery-less JACE scenarios” on page 2
 - “Battery-less versus battery trade-offs” on page 2
- “SRAM plus battery scenarios” on page 3
- “Unsuitable station for SRAM support” on page 4
 - “Unsuitable station example” on page 4
 - “Second example station - suitable for SRAM support?” on page 4
 - “Disabling SRAM support” on page 5
- “Requirements for JACE controller SRAM support” on page 5
 - “Licensing” on page 5
- “How SRAM support in a JACE works” on page 5
 - “Usage goal” on page 5
 - “SRAM based solution” on page 5
 - “Operation overview” on page 6
- “About the DataRecoveryService” on page 8
 - “Data Recovery Service Editor” on page 8
 - “DataRecoveryService properties” on page 10
- “Diagnostic tools for the DataRecoveryService” on page 11
 - “Station output from the DataRecoveryService” on page 11
 - “Spy page for the DataRecoveryService” on page 12
 - “Reformatting SRAM” on page 12
- “Document change log” on page 13

SRAM and/or battery options

With any SRAM-equipped controller, including the “onboard SRAM” JACE controllers like the JACE-6E and JACE-3E, there are two *different* ways to utilize SRAM. There is also a scenario in which you may elect *not* to make use of SRAM, even if the controller includes it standard (onboard).

Battery-less JACE (with SRAM)	JACE with SRAM plus backup battery	JACE using <i>only</i> backup battery
JACE-6E/JACE-3E (SRAM card unnecessary) 	JACE-6E/JACE-3E (SRAM card unnecessary) 	JACE-6E/JACE-3E or JACE-2/6 series 
The “battery-less” controller was the original NiagaraAX implementation of SRAM. No backup battery (NiMH or otherwise) is attached to the JACE. For more details see “Battery-less JACE scenarios” on page 2.	Since 3.6.44 or later, an SRAM-equipped JACE can utilize <i>both</i> SRAM and an installed backup battery, like the NiMH battery pack shown above. For more details see “SRAM plus battery scenarios” on page 3.	Sometimes a station is a poor candidate for SRAM support. Disabling SRAM operation is best in this case, even if “onboard” SRAM. For related details, see “ Unsuitable station for SRAM support ” on page 4.

Battery-less JACE scenarios

A standard JACE-6E or JACE-3E controller, or any SRAM-equipped JACE (via SRAM option card), when installed *without* an integral, rechargeable battery pack (or external 12V battery), provides a “battery-less” installation. This can offers advantages in certain situations. Some example scenarios are:

- Installation of any battery-equipped device is *forbidden*, due to job site or local regulations.
- Installation without a battery may allow a *higher temperature environment rating*, such as a JACE-6E or JACE-3E (60 °C maximum, vs. 50 °C maximum if with a rechargeable NiMH battery pack).
- Over time, the periodic replacement of the NiMH battery in the JACE presents too many costly obstacles—for whatever reason. For example, a JACE may be installed in a difficult-to-reach location. And as the NiMH battery’s condition declines, “battery bad” alarm notifications are received.

You can meet such issues by installing a standard JACE-6E or JACE-3E controller, or if another earlier model (e.g. JACE-2,-6,-7), by removing its NiMH battery pack and installing the SRAM option card, makes the unit “battery-less”. As a consequence of SRAM-only backup support, JACE monitoring of batteries no longer occurs—no more “battery bad” alarm notifications.

Note: A station running in a battery-less JACE has no seamless immunity to “power bumps”. Although all station data, including components, histories, and alarms, are automatically restored to “pre-event” values as part of station startup (following power restoration), the briefest power outage results in a controller reboot. For more details, see “[Battery-less versus battery trade-offs](#)” on page 2.

Note that a “battery-less” configuration for an SRAM-equipped controller is now *one of two* possible configurations where SRAM is used. For more details, see “[SRAM plus battery scenarios](#)” on page 3.

Battery-less versus battery trade-offs

Any JACE controller with a charged backup battery holds a *key advantage* over a battery-less SRAM-equipped JACE, in this regard:

- Station operation continues (uninterrupted) across very short power outages, that is “power bumps” lasting only a few seconds—*without* initiating an orderly shutdown.
 [A UPS could be used to mitigate this, but this would re-introduce battery maintenance for the UPS.]

The NiMH battery provides enough power for this, immediately recharging when power is restored. If a power outage lasts longer than its defined “shutdown delay” time, the NiMH battery allows sufficient time for the JACE to perform an “orderly shutdown”, including saving the station’s database (config.bog) and all recorded alarm and history records.

However, a JACE without SRAM but with a *weak* NiMH battery is exposed to a different power outage issue—where the potential exists for data loss (since the last station save), due to insufficient battery power to complete an orderly shutdown. Potentially, this could also occur if enough power outages occur in rapid succession—draining the controller’s battery to a low level.

A battery-less JACE solves that problem, as all station-generated data (changed from that stored in its non-volatile flash memory at the time of power loss) is always preserved in SRAM. Upon power restoration, this data is “played back” in the station during startup, then saved in its non-volatile flash memory.

SRAM does not preserve data or files external to station

Please note that if a JACE power event occurs when station users have unsaved file changes, say in a Px file or Nav file being edited, those unsaved changes *are lost*. This behavior may *seem* different from a battery-equipped JACE entering an ordered shutdown—but it is not.

The practical difference is that a battery-equipped JACE may keep running over a short “power bump”. Station users may be aware of such an event, and react by saving changes (click **Save** button in the active view). Providing that communications are still established, the file edited may be saved. Or, power may be lost only momentarily, and then remain stable until the user does a normal save.

Note a battery-less SRAM-equipped JACE does not provide a similar save opportunity after a power bump—it is already busy rebooting. Therefore, as a best practice, you should advise system users of battery-less SRAM-equipped JACEs to save often when editing items like Px graphics and Nav files.

SRAM plus battery scenarios

An SRAM-equipped controller can utilize both its SRAM *and* an installed backup battery, such as a JACE-6E or JACE-3E controller with an *optional* internal NiMH battery pack, or a JACE-2,-6,-7 controller (*with* SRAM option card) and its standard NiMH battery pack. This also applies to a QNX-based JACE controller with an external 12V sealed lead-acid battery, either as its main backup battery (JACE-603, JACE-645) or as a battery in “parallel” with its NiMH battery (JACE-7).

This hybrid “SRAM plus battery” configuration offers:

- Immunity to power bumps, thanks to the backup battery—similar to a unit with backup battery only.
- A longer allowable “Shutdown Delay” time, up to 10 minutes for a unit with only a NiMH backup battery. This is beyond the 30 seconds (recently 1 minute) maximum period. After continuous operation this long on battery power, at shutdown the JACE saves its database and powers off. So, a unit with a reasonably healthy NiMH backup battery could continue running over a power outage of up to 10 minutes. Note that the station’s “DataRecoveryService” must be enabled to specify this longer shutdown delay in station’s “PowerMonitorService”. If the unit is equipped with a sealed lead-acid (SLA) battery, this shutdown delay can be specified up to 15 minutes.
- Data independence from backup battery condition, again a consequence of the ongoing SRAM operation. If the charge on the backup battery weakened such that the controller was unable to complete a database save on a controlled shutdown, or even if immunity to power bumps was lost, data integrity still remains. Upon reboot from a power restoration, the station’s DataRecoveryService restores (replays) the previously recorded station runtime data from SRAM.

Note a discharged battery could happen if multiple, consecutive power outages occurred, draining the charge on the rechargeable backup battery. A longer “Shutdown Delay” time *could* contribute to this. However, unlike with a controller not using SRAM, station runtime data is safe in this case.

Gain the features above by installing an optional NiMH battery pack in a standard JACE-6E or JACE-3E, or if another earlier model (e.g. JACE-2,-6,-7), by *retaining* its NiMH battery pack when installing the SRAM option card. By default, this hybrid configuration is standard with a “retrofit board” JACE-603 or JACE-645, assuming the controller’s 12V SLA battery (in the controller’s enclosure) is retained.

Note in this hybrid configuration, JACE monitoring of backup battery(ies) continues—so “battery bad” alarm notifications are still possible, and the regular replacement of backup batteries is still needed.

Again, the “SRAM plus battery” configuration for an SRAM-equipped controller is *one of two* possible configurations where SRAM is used: the other is for a JACE controller to be “battery-less”. See “[Battery-less JACE scenarios](#)” on page 2.

Note: *Although this hybrid configuration is typically the most desirable, note that some stations may be poor candidates for SRAM operation, with SRAM support even counter-productive. In this case, a controller platform that includes “built-in” SRAM (such as a JACE-6E or JACE-3E) can be configured to disable its DataRecoveryService, and use only its (optional) installed backup battery. Or, an SRAM option card can be removed from another JACE controller, such that it also uses only its installed backup battery. For more details, see “[Unsuitable station for SRAM support](#)” on page 4.*

Unsuitable station for SRAM support

To summarize SRAM operation, the DataRecoveryService writes values to SRAM as they occur. SRAM is divided into 3 separate blocks. When a block is filled, it is copied from SRAM to the controller's flash memory. During this "flush process," data is written to a different block in SRAM.

By default, the DataRecoveryService sets a threshold of 10% of the free flash disk space on the JACE controller for SRAM block data. Once the number of blocks exceeds this threshold, a station save is forced. When the station save successfully completes, those blocks are no longer needed and are deleted from the flash memory.

With this in mind, note that in some cases it is possible a JACE's station may be an unsuitable candidate for SRAM support. For example, a station with many "change of value" (COV) histories, associated with rapidly changing values, may result in SRAM data buffers filling too often. This results in the "automatic database save," as performed by the DataRecoveryService when SRAM buffers become full, to occur too frequently—possibly every couple of minutes.

Ideally, such database saves (to flash memory) should occur no more than once an hour.

High frequency database saves are undesirable for two reasons:

- Inefficient use of the JACE's CPU time, as each automatic save consumes processing power otherwise used by other processes (threads) in the station.
- Possible flash memory problems in the future, resulting from too many writes every few minutes—or minute. Even though the file system on the JACE performs flash-wear leveling, excessive writes to flash could lead to problems over time.

See the following station examples:

- [Unsuitable station example](#)
- [Second example station - suitable for SRAM support?](#)

Unsuitable station example

An example test station running in a JACE-202 Express was found to fit this unsuitable category. This station had 176 COV histories, each writing every 5 seconds, along with other NRIO components and BACnet components (making little additional history usage). SRAM support in this station resulted in an ongoing automatic save approximately *every 6 minutes*—about a magnitude away from ideal.

176 COV histories changing every 5 seconds yields 126,720 changes/hour ($176 * 12/\text{min} * 60 \text{ min/hr}$).

Second example station - suitable for SRAM support?

Now consider the suitability of a station with 1200 *interval* histories, each with a collection interval of 10 to 15 minutes. Planning for the "biggest load" configuration, that is all histories at 10 minute intervals:

1200 histories changing every 10 minutes yields 7,200 changes/hour ($1200 * 6/\text{hr} * 60 \text{ min/hr}$).

Note this example station results in a rate of history changes nearly *18 times less* than the first ([Unsuitable station example](#)), where the ratio of these two "total changes per hour" is 17.6 ($126,720/7,200$).

Let's assume everything else is equal between this station and the first (unsuitable) example, including the same amount of free flash space. Then in this case, ongoing automatic database saves would occur approximately *every 1.5 hours* ($6 \text{ minutes} * 17.6$, or 105 minutes)—well inside the "ideal range".

Note: *A number of different variables factor into the actual operation of SRAM support, including:*

- Rate of changes that need to be persisted.
- Types of changes (histories, alarms, and setpoint changes all have different sizes).
- Amount of free flash disk space on the controller.

Therefore, the examples above do not reflect a full range of different scenarios.

Also see the following related sections:

- The next section, "[Disabling SRAM support](#)".
- For detailed background information on the operation of SRAM support, see "[How SRAM support in a JACE works](#)" on page 5.
- For tools to monitor operation, see "[Diagnostic tools for the DataRecoveryService](#)" on page 11.

Disabling SRAM support

In situations where a station is not compatible with SRAM support (see “[Unsuitable station for SRAM support](#)”), you can equip its host JACE controller with a suitable backup battery and disable SRAM operation. Depending on the controller series, do this in one of several ways:

- If a JACE-6E or JACE-3E, install the optional NiMH battery pack. In its station’s **DataRecoveryService**, set the “Service Enabled” property to `false`. See “[Data Recovery Settings](#)” on page 8. In the station’s **PlatformServices** container, make sure the “Battery Present” property is true.
- If a JACE platform using an SRAM option card, remove that option card and install its NiMH battery pack. Then using the platform **Software Manager**, uninstall the `platDataRecovery` module to remove the **DataRecoveryService** from the station. Again, in the station’s **PlatformServices** container, make sure the “Battery Present” property is true.

Requirements for JACE controller SRAM support

The following is required for SRAM support in a JACE controller, using either “onboard” SRAM or the SRAM option card, along with the platform `DataRecoveryService`:

- A JACE-6E or JACE-3E controller with onboard SRAM, *or*
A JACE-603 or JACE-645 with onboard SRAM, licensed to run a NiagaraAX station, *or*
A JACE-2,-6,-7 controller running AX-3.6 or later, with an available option card slot, *and*
 - An SRAM option card, installed in that option card slot.
 - The “`platDataRecovery`” module must be installed in the JACE (this module is automatically installed in platforms that do not need the SRAM option card).
- The JACE must also be licensed for this feature—see “[Licensing](#)”.

Licensing

The JACE needs a license entry as below, to support the SRAM option card and `DataRecoveryService`:

```
<feature name="dataRecovery" expiration="never" parts="partName" />
```

If the `dataRecovery` license feature is missing, the installed SRAM option card will not work—the `DataRecoveryService` in its station will either be missing, or will otherwise remain in a fault condition.

How SRAM support in a JACE works

This section explains how SRAM support in a JACE controller works, with the following sections:

- [Usage goal](#)
- [SRAM based solution](#)
- [Operation overview](#)

Usage goal

When recovering from a power outage (where a “controlled shutdown” did *not* occur), the goal is to retain all runtime station data in volatile DRAM (Dynamic RAM) that had changed prior to power lost, that is, since the last station save to the JACE’s NVRAM, also known as “flash” memory. Changes apply to all normally persisted station data, including any changes to components, histories, and alarms.

Simply increasing the flash write frequency (for example upon each value change) is not viable because of eventual damage to flash memory components. Additionally, the latency of flash writes is significant; such writes may not complete if power is lost.

Replacing all flash memory on a JACE with SRAM is cost prohibitive; however, a memory caching scheme using SRAM was developed. See the section “[SRAM based solution](#)”.

SRAM based solution

The patent-pending SRAM support feature employs both hardware and software, as follows:

- Hardware: SRAM buffer with dedicated capacitor for memory refresh (backup). Newer JACE-3E and NPM6E-based controllers (JACE-6E, JACE-603, JACE-645) have “onboard” 512 KB of SRAM and backup capacitor, plus supporting circuitry. The SRAM option card provides all of this on an installable option card, for JACE-2,-6,-7 series controllers.
- Software: The JACE requires AX-3.6 or later and the module “`platDataRecovery`”, which works with the SRAM buffer to continuously save changes as “deltas”. Essentially, this feature automatically “records” all changes during runtime, for “playback” upon station startup after a power outage.

Providing that the JACE is properly licensed, no other configuration is required for this combination to provide SRAM support. This feature automatically works for a JACE-6E or JACE-3E, as well as for other platforms with the SRAM option card (providing the platDataRecovery software module is installed).

However, an interface “window” is automatically provided in the station running on the JACE, via the dynamically created “DataRecoveryService” under its PlatformServices container. A special view on this platform service allows verification of operation, and if ever necessary, a means of troubleshooting.

Note: *Starting in build 3.6.44, the DataRecoveryService includes a single configuration property: “Service Enabled”, which is true by default. The use case is for a JACE-3E or NPM6E-based controller (e.g. JACE-6E) in which you want to disable SRAM support, using only its installed backup battery instead. For background details, see “Unsuitable station for SRAM support” on page 4.*

Operation overview

Providing the JACE controller is equipped with SRAM *and* is properly licensed, the platDataRecovery module dynamically creates a “DataRecoveryService” in the running station’s PlatformServices container. This service presides over the SRAM option card, which in turn acts as a buffer for flash memory.

The new service partitions the SRAM into multiple buffers, or SRAM data “blocks”. In the initial usage of SRAM, there are 3 blocks. Only *one* these SRAM blocks is ever “active” at any time.

Operation can be described in the following modes:

- [Recording \(normal operation\)](#)
- [Station save effects](#)
- [Playback scenario \(power lost or reboot\)](#)

Recording (normal operation)

Immediately after a station has started, any subsequent change in any of its three object spaces (component, alarm, history) is immediately “recorded” by the DataRecoveryService, writing to the currently active SRAM block. In bytes, each block has a “total capacity”, a calculated amount of needed “overhead”, and the current available “free” area. Exceptions to this recording are noted below.

- Unsaved user edits to *files*, e.g. Nav or Px files (working in the Px Editor) are not recorded, and thus are lost upon any power event if running battery-less. See [“SRAM does not preserve data or files external to station”](#) on page 3.
- Slots of objects that have the “Non-Critical” config flag set are also not recorded. Note by default, this new slot config flag (AX-3.6 and later) is cleared. See [“Non-Critical config flag”](#).

When the active SRAM block becomes full, i.e. its “free” area is not large enough for a record write, its data is written out (“flushed”) to a file in the JACE’s flash memory, then that SRAM block is cleared. Concurrently, while this transfer/flush occurs, another SRAM block is activated and used. This multi-threaded buffer method enables recording station changes that would otherwise be lost or blocked.

Over time, many SRAM blocks may fill, where each results in another file in flash on the JACE. These files are automatically named in a numeric sequence with a “.drdb” extension (data recovery database), stored in the station’s file space under the !\dataRecovery directory. For example, this folder may contain files “1.drdb”, “2.drdb”, “3.drdb”, and so on, with “1.drdb” being the oldest. This sequence becomes important later, when the DataRecoveryService restores (plays back) data.

At some point, the collective size of the station’s “.drdb” files may exceed the “persistent capacity” limit used by the DataRecoveryService. This is a “persistent space full” condition. If this occurs, an automatic *station save* occurs. Effects on the DataRecoveryService are the same as if a station save is manually issued, or if an automatic save from the platform “auto-save” frequency occurs. See [“Station save effects”](#).

Non-Critical config flag There may be cases where new slots or the data values of those slots is not considered critical, and does not need to be saved by the DataRecoveryService. To accommodate for this, a new “Non-Critical” config flag can be set on *any slot* (AX-3.6 and later). By default, this flag is cleared, meaning it is *not* set.

When this flag is set on newly-created slots, such new slots are *not* saved by the DataRecoveryService. If running “battery-less” and power is lost before the station is saved, these slots will be permanently *lost*.

However, if power is not lost, then the slots are still saved as part of a routine station save. But when the values of the slots change, those changes are not recorded by the DataRecoveryService; they are only saved when a “station save” occurs. Not saving this non-critical data through the DataRecoveryService may offer some performance advantages.

Station save effects

When saving a station running the DataRecoveryService, the following things occur. Note this applies regardless of how the save was issued—for example, a manually invoked command (**Save Station**), or an automatically issued save, e.g. a “persistent space full” condition by the DataRecoveryService, or a station save that occurs as part of a station copy operation, reboot command, or from a “controlled shutdown” while running on backup battery power:

1. The normal station save method is used to capture all changes to all the station’s object spaces (components, alarm, history) to flash memory, saved as the file `config.bog`.
2. Upon a *successful save*, all “.drdb” files holding buffered data are *erased from flash*, as these are no longer necessary.
3. The active SRAM data block is also cleared; however, one block operates in a “reserved state”. This is needed to capture any changes that may occur while the station is in the “saving” state. Otherwise, data loss could potentially occur if power was lost during or immediately after a save.

Again, note that buffered data in flash and SRAM is not erased *until* the save (`config.bog`) is successfully written to flash. Thus, data recovery records are not lost if a power loss occurs during a save.

Playback scenario (power lost or reboot)

- If power is lost, a battery-less JACE is immediately off, with the last active SRAM block preserved by capacitor charge, plus typically one or more flash-preserved “.drdb” files. This could also occur if battery-equipped controller was unable to successfully complete an orderly shutdown. The JACE remains in this dormant state until power is restored, when the JACE boot process begins normally.
- If a reboot command is given, the last active SRAM block is present, but typically no flash-preserved “.drdb” files—because a reboot typically¹ first issues an “orderly shutdown”, starting with a station save. See “[Station save effects](#)”. The JACE boot process begins normally.

During station startup, after loading into DRAM the station’s last saved database (`config.bog`), the DataRecoveryService starts up and detects that recovered data exists in the preserved SRAM block, and also possibly in flash-preserved “.drdb” files.

The DataRecoveryService then performs the following actions:

1. It replays (writes to DRAM) all the flash files in sequence, beginning with the oldest.
2. Once flash records have been replayed, the service replays the records found in the SRAM block, as they are the most recent.

Note that in some cases this playback to DRAM can take a few minutes—this depends on how much data there is to replay. Station startup then completes, and an *immediate save* of the DRAM-restored station to flash (`config.bog`) occurs. All previous “.drdb” files are erased in flash, as well as SRAM data blocks, and DataRecoveryService operation begins anew. See “[Recording \(normal operation\)](#)” on page 6.

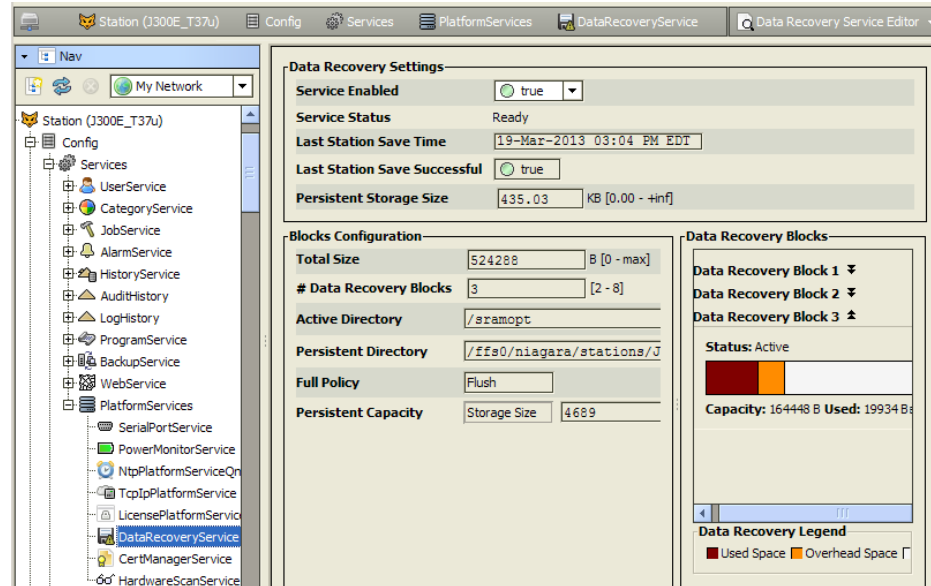
1. A reboot issued from system shell is one exception—no orderly shutdown occurs before such a reboot. There may be .drdb files saved when this reboot happens. If so, these are “replayed” the same as if power is lost.

About the DataRecoveryService

The DataRecoveryService is the station platform service that provides SRAM support for any JACE equipped with SRAM. Providing the `platDataRecovery` module is installed, this service automatically appears under PlatformServices (Figure 1).

- If the JACE is not properly licensed, this service remains in fault. See “[Licensing](#)” on page 5.
- This service includes a “Service Enabled” configuration property. The DataRecoveryService does *not* replace the PowerMonitorService, as in the initial AX 3.6 release.

Figure 1 Data Recovery Service Editor in PlatformServices of SRAM-equipped JACE



The figure above shows the default view for the service: the **Data Recovery Service Editor**. By default, this view is “informational” only—all but one slot in the service are (read-only) status—*no configuration is needed* (unless disabling SRAM support, and using only an installed backup battery).

Note: For background details about its operation, see “[How SRAM support in a JACE works](#)” on page 5.

Note the Figure 1 example reflects a scenario where a station save has occurred since the service was created. Some SRAM “data recovery blocks” have already been flushed to flash (“Persistent Storage Size” is not 0.00 KB). For related details, see “[Data Recovery Service Editor](#)”.

Note alarm-related properties are in a different view—see “[DataRecoveryService properties](#)” on page 10.

Data Recovery Service Editor

This default view of the DataRecoveryService has three main areas, described as follows:

- [Data Recovery Settings](#)
- [Blocks Configuration](#)
- [Data Recovery Blocks](#)

Data Recovery Settings

All but the first of these is a read-only status property, and include the following:

- **Service Enabled**
Defaults to `true`, to enable SRAM support via this service. If a JACE with onboard SRAM (e.g. JACE-6E or JACE-3E), you can set this to `false` to *disable* SRAM support—in which case it relies on its installed backup battery and its **PowerMonitorService** to preserve station data upon loss of power. (If an SRAM option card-equipped JACE, you can simply *remove* the SRAM option card and *uninstall* the `platDataRecovery` software module).
- **Service Status**
The current status of the DataRecoveryService, which is typically “Ready”. Other states include “Starting”, “Configuring”, “Replaying”, “Saving”, “Stopping”, “Stopped”, “Fault” and “Unknown”.
- **Last Station Save Time**
Reflects the last time a station save occurred (`config.bog` written to flash memory). This save may (or may not) have occurred as a result of the DataRecoveryService.

- **Last Station Save Successful**
Boolean that reflects if last station save attempt was successful, as either “true” or “false”. This save may (or may not) have occurred as a result of the DataRecoveryService. Note in the case of a newly-created DataRecoveryService, this is “false” until the next save occurs.
- **Persistent Storage Size**
Reflects the total size of all the “flushed to flash” data block files (“.drdb” files) that exist in the station’s !/dataRecovery folder, in KB. Initially, this will be 0, until the first SRAM block flushes to flash. It will then increment by that KB amount for each subsequent SRAM block flushed. Note this value is continually compared to the “Persistent Capacity” property in the [Blocks Configuration](#) property section.

Blocks Configuration

These status properties include the following:

- **Total Size**
Reflects, in bytes, the total amount of SRAM buffer memory available to the service. For example, this is “524288” for the 512 KB SRAM option card.
- **Number of Data Recovery Blocks**
Reflects the number of data block partitions of SRAM used, for example, 3.
- **Active Directory**
Reflects the directory used in SRAM for the active data block.
- **Persistent Directory**
Reflects the full flash file directory path used to store flushed “.drdb” files, which equates to: !/dataRecovery
- **Full Policy**
Reflects the current policy when an SRAM data block becomes full (currently “Flush”).
- **Persistent Capacity**
Reflects the size limit, in KB, for the total of all “flushed to flash” data block files (“.drdb” files). If this limit is exceeded (see property “Persistent Storage Size”), the service automatically triggers a *station save* operation. For related details see [“Station save effects”](#) on page 7.

Data Recovery Blocks

This area provides expandable bar graphs for each of the SRAM buffer data blocks, to visually represent the current amount of used space, overhead space, and available free space, along with numerical values.

By default, the currently active SRAM block is expanded, showing a bar graph of current buffer usage.

Figure 2 Example SRAM data block, show both states “Active” and “Flushing”

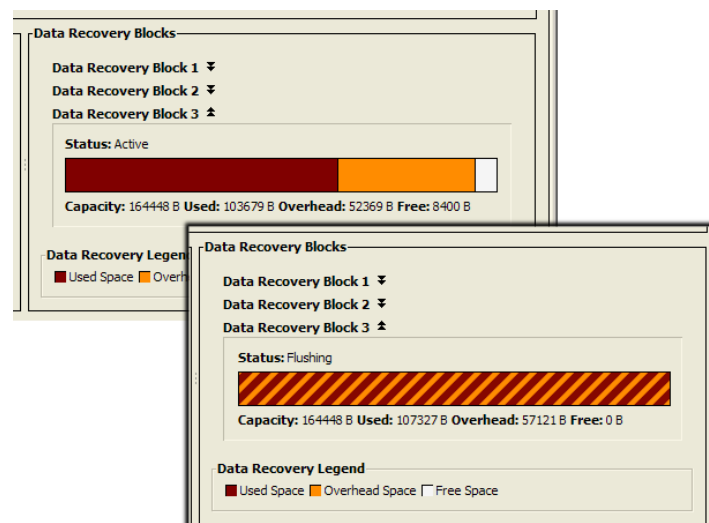


Figure 2 shows an example active SRAM block first near full, then “flushing” momentarily to flash—an operation that lasts only a second or two. Another SRAM block becomes active when this happens, and it is used until it fills and needs to flush to flash.

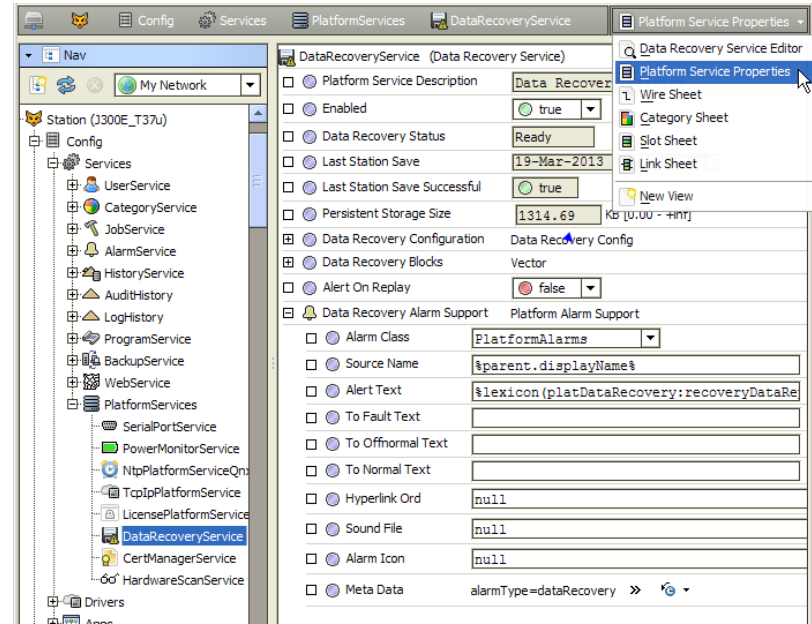
Above the bar graph of each block, its **Status** is described, typically as either: “Active”, “Idle”, or sometimes “Flushing”, with other states “Purging”, “Awaiting Idle”, “Flush Queued”, “Defragmenting”, “Reserved”, “Fail”, and “Unknown”.

Below the bar graph of each block, numerical amounts display, in bytes, for its total Capacity, currently Used space, calculated Overhead Space, and available Free Space.

DataRecoveryService properties

In addition to the (default) **Data Recovery Service Editor** view, the **DataRecoveryService** also has properties on its **Platform Service Properties** view, see [Figure 3](#).

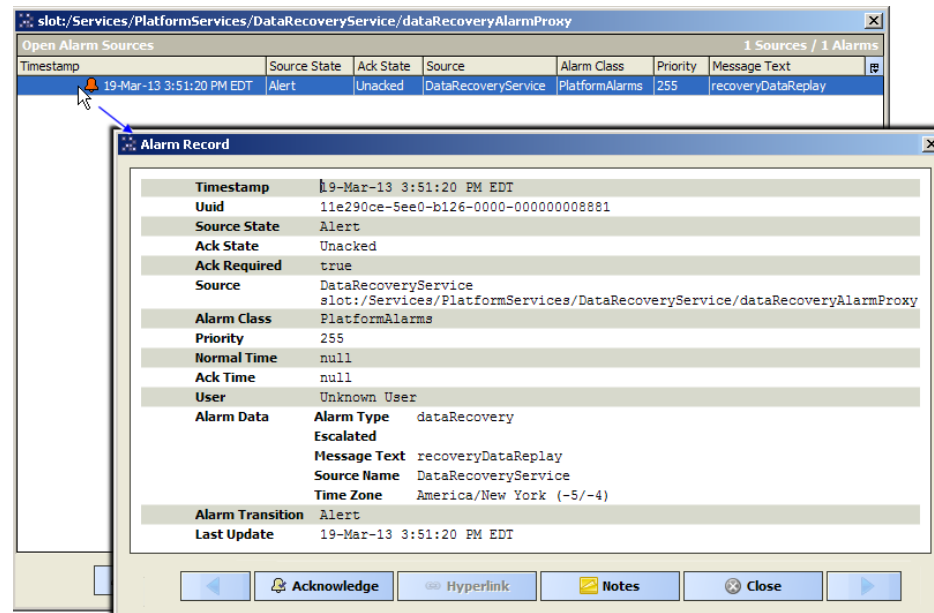
Figure 3 Platform Service Properties view of DataRecoveryService (for alert/alarm routing)



Most of these properties are also on the **Data Recovery Service Editor** default view (see [Figure 1](#) on page 8). However, the following alarm-related *configuration* properties are accessible only via this view:

- **Alert On Replay**
Default is *false*. If set to *true*, upon any controller boot sequence in which SRAM recorded data is discovered and played back, a corresponding *alert* is routed to the Alarm Class named in the **Data Recovery Alarm Support** container. [Figure 4](#) shows details for such an example alert.

Figure 4 Example Alarm Record details for an alert generated by DataRecoveryService



- **Data Recover Alarm Support**

This is the standard container slot for routing platform service-generated alarms or alerts, in this case an *alert* from the **DataRecoveryService** upon any controller boot sequence in which SRAM recorded data is discovered and played back. These properties work in the same fashion as those in an alarm extension for any control point.

Diagnostic tools for the DataRecoveryService

The **DataRecoveryService** provides diagnostic tools that may be useful if troubleshooting, as follows:

- [Station output from the DataRecoveryService](#)
- [Spy page for the DataRecoveryService](#)

A procedure for reformatting the SRAM option card (if necessary) is given. See [“Reformatting SRAM”](#) on page 12.

Station output from the DataRecoveryService

When left at the default “message” log level, the DataRecoveryService produces minimal station output related to operation. Primarily, related messages are seen at station startup, especially if following a power loss event or reboot command.

```
MESSAGE [11:18:15 13-Aug-10 EDT] [sys.registry] Up-to-date [250ms]
MESSAGE [11:18:20 13-Aug-10 EDT] [sys.registry] Loaded [2935ms]
MESSAGE [11:18:29 13-Aug-10 EDT] [sys] Baja runtime booted ("/ffs0/niagara")
MESSAGE [11:18:29 13-Aug-10 EDT] [sys] Loading "/ffs0/niagara/stations/J202_TestW/config.bog"...
MESSAGE [11:19:34 13-Aug-10 EDT] [sys] Loaded (64809ms)
MESSAGE [11:19:56 13-Aug-10 EDT] [alarm.database] Loading...
MESSAGE [11:19:58 13-Aug-10 EDT] [alarm.database] Loaded [2196ms, 32 alarms, 104 pages]
WARNING [11:19:58 13-Aug-10 EDT] [platDataRecovery.service] data recovery detected, replaying...
MESSAGE [11:20:11 13-Aug-10 EDT] [sys] DataRecoveryService restoration check complete (18368ms)
MESSAGE [11:20:12 13-Aug-10 EDT] [sys] Services Initialized (1010ms)
MESSAGE [11:20:12 13-Aug-10 EDT] [sys.mixin] Updated [112ms]
MESSAGE [11:20:14 13-Aug-10 EDT] [history.db] Starting async warmup of history config index...
MESSAGE [11:20:14 13-Aug-10 EDT] [history.db] Async history config index warmup completed in 342 ms.
MESSAGE [11:20:37 13-Aug-10 EDT] [web.server] HTTP Server started on port 80
MESSAGE [11:20:38 13-Aug-10 EDT] [fox] Service started on port 1911
MESSAGE [11:20:39 13-Aug-10 EDT] [sys] Niagara Runtime Environment: 3.6.20
MESSAGE [11:20:39 13-Aug-10 EDT] [sys] *** Station Started (27273ms) [153291ms total] ***
niagara>MESSAGE [11:20:41 13-Aug-10 EDT] [sys] Saving station...
MESSAGE [11:20:55 13-Aug-10 EDT] [history.db] Saved history archive (4197ms)
MESSAGE [11:21:02 13-Aug-10 EDT] [sys] Saved /ffs0/niagara/stations/J202_TestW/config.bog (19654ms)
```

The example above shows related messages mixed in with other station startup entries.

Using “spy”, you can increase the station’s log level, by setting it to “trace” on items:

- platDataRecovery.manager
- platDataRecovery.service

This produces *much* more debug information in station output. An example small snippet below reflects trace level output when an SRAM data block “flushes” to a flash file.



Caution

Turning trace logging on either of the logs above may produce extremely large levels of output, and should not be left in this log level state. This applies particularly to large stations.

```
TRACE [10:31:12 13-Aug-10 EDT] [platDataRecovery.service] External append(history:, encoded key
bytes: appended /J202_TestW/AuditHistory...
TRACE [10:31:12 13-Aug-10 EDT] [platDataRecovery.manager] Size of used block exceeds free
space, forcing flush of block
TRACE [10:31:12 13-Aug-10 EDT] [platDataRecovery.service] Narcissistic append(encoded key
bytes: AA==, data bytes: 0000b10b000e...
TRACE [10:31:12 13-Aug-10 EDT] [platDataRecovery.service] Attempt to append data with key
encoded as bytes: AA== successful
TRACE [10:31:12 13-Aug-10 EDT] [platDataRecovery.service] Flush operation successful.
TRACE [10:31:12 13-Aug-10 EDT] [platDataRecovery.service] Attempt to append data with key
encoded as bytes: appended /J202_TestW/AuditHistory...
```

You can also increase the log level of other items related to (or used by) the DataRecoveryService, including the following:

- alarm.dataRecovery — For logs about the AlarmService’s use of the DataRecoveryService
- history.critical — For logs about the HistoryService’s use of the DataRecoveryService
- As well as these additional DataRecoveryService related items:
- sys.critical
- sys.critical.load.item
(item = add, change, facetsChange, flagsChange, recategorize, remove, rename, reorder)
- sys.critical.progObj
- sys.critical.save.item

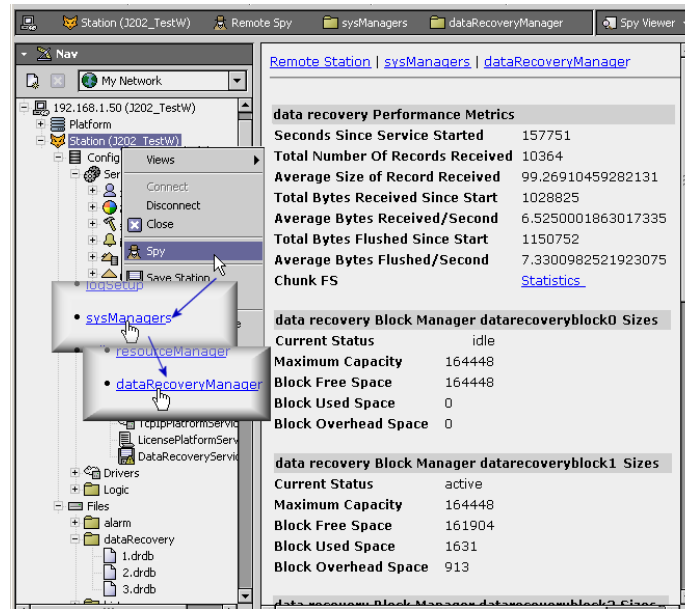
(item = add, change, facetsChange, flagsChange, recategorize, remove, rename, reorder)

The same caution applies as before about trace level on these items—some may produce very large levels of station output.

Spy page for the DataRecoveryService

An extensive “spy” page on a station’s [DataRecoveryService](#) is available in Workbench at the following location: **Spy > sysManagers > DataRecoveryManager**

Figure 5 Spy page for DataRecoveryService



This spy page, partially shown in [Figure 5](#), provides numerous statistics about station data being backed up by the service. Details about these statistics are outside the scope of this document.

Reformatting SRAM

When using a “serial shell” connection to a JACE controller with SRAM, a special “alternative boot” option is available. A sub-option allows you to reformat the onboard SRAM (or SRAM option card) installed in the JACE.

You must be near the JACE to make this direct serial shell connection, and also be able to easily power it off and on as needed.

Note: *In most cases the following reformat procedure will never be necessary. When shipped, the SRAM option card is already properly formatted, and does not contain any stored data.*

However, in rare cases it may be necessary to reformat the SRAM option card using this procedure.

For details about making a serial shell connection, see “System shell” in the *JACE NiagaraAX Install and Startup Guide*.

To reformat SRAM option card in a JACE

- Step 1 If needed beforehand, backup the JACE to your PC using normal Workbench platform tools.
- Step 2 Power the JACE *off*, and install the serial shell jumper (if not already installed).
Connect the necessary serial cable between the JACE and your PC’s COM port.
- Step 3 On your PC, start a terminal emulation program, for example HyperTerminal, and open a previously saved setup for JACE-2,-6,-7 communications, using that COM port. Settings are listed below.
 - Bits per second: 115200
 - Data bits: 8
 - Parity: None
 - Stop bits: 1
 - Flow control: Hardware
- Step 4 With your HyperTerminal session active, apply power to the JACE.
After some number of seconds, text should appear in the HyperTerminal window similar to below:

Note: Wait for the prompt to “Press ESC to choose alternate boot options...”, then press ESC.

```
IPL for NPM 2 (PPC405EP) v3.03 ECC
Press <ctrl-c> to stop autoboot...
Autobooting...
Loading image from on-board nand flash
npm2xx startup version 6.41
launching devc
waiting for /dev/ser1
creating symlink for /dev/serconsole
Welcome to QNX Neutrino 6.4 on the NPM 2xx (ppc405)
starting /ffs0 filesystem
ETFS_FS_512

Press ESC to choose alternate boot options...
```

Step 5 Press ESC to go to the alternate boot menu, a *portion* of which is shown below.

Note: As explained in the alternate boot menu, be careful about which options you skip!

```
9. Skip starting niagara daemon
10. Skip SRAM option card mount
11. Format SRAM option card before mount

c. continue with boot

Enter choice:
```

Step 6 At the “Enter choice” prompt, type 11 and press ENTER.

The alternate boot menu repeats, only *missing* the selected item 11 (“Format SRAM option card before mount”).

Step 7 Type `c` (to continue with boot) and press ENTER

You see “continuing boot...” along with other normal boot messages for TCP/IP initialization.

When the boot completes (at the login prompt), there should be entries indicating that a format operation occurred, as shown in the following:

```
SRAM option card detected in slot 1
Formating SRAM option card before mount
SRAM option card mounted successfully at /sramopt
starting ntpd...
MESSAGE [15:36:48 29-Jun-2010] [tid=1] niagarad: starting, baja_home=/niagara
MESSAGE [15:36:48 29-Jun-2010] [tid=1] webserver: web server started [tid = 4]
MESSAGE [15:36:48 29-Jun-2010] [tid=1] app registry: station registry starting
MESSAGE [15:36:48 29-Jun-2010] [tid=7] engine watchdog: app J202_TestW watchdog
thread started [tid = 8]
MESSAGE [15:36:48 29-Jun-2010] [tid=7] station: station J202_TestW starting
login:
```

Step 8 When the “login” prompt appears, power off the JACE.

If needed (and applicable), you can now remove the SRAM option card to use in another JACE controller.

Note: If removing an SRAM option card from a JACE-2,-6,-7 controller and not reinstalling, be sure to uninstall the `platDataRecovery` module from the JACE controller, using the platform **Software Manager**.

Step 9 Exit from the HyperTerminal application, selecting to **Save** if you wish to reuse this setup again.

Document change log

Updates (changes/additions) to this *JACE Data Recovery Service (SRAM support)* Engineering Notes document are listed below.

- Updated: March 25, 2013
Updated to mention another JACE hardware platform with *onboard* SRAM (JACE-3E controller), introduced near the time of an AX-3.7 update release (build 3.7.105 or later). Like the JACE-6E controller, the JACE-3E ships without a NiMH backup battery, but you can optionally install one if desired. A couple of document sections applied only if using the `DataRecoveryService` in the initial release of AX-3.6, and so were subsequently removed. Most other document sections remained unchanged, but a bit more information was added to “[Unsuitable station for SRAM support](#)” on page 4. Several screen captures were updated to show a JACE running AX-3.7.
- Updated: February 9, 2012
Updates to describe changed operation of the `DataRecoveryService` starting in build 3.6.44 or later, coinciding with new JACE hardware platforms with *onboard* SRAM (JACE-6E, JACE-603, JACE-

645). Now any SRAM-equipped JACE controller (including one with an SRAM option card) can also utilize an installed backup battery, if desired.

Most document sections were reworked and some *new* sections added, including “SRAM and/or battery options” on page 2, “SRAM plus battery scenarios” on page 3, and “Disabling SRAM support” on page 5. The section “About the DataRecoveryService” on page 8 now explains the service in build 3.6.44 or later, and includes a new subsection “DataRecoveryService properties” on page 10.

- Publication: February 10, 2011
Initial document.