

Technical Document

Sedona Framework Sox Tools Guide

November 7, 2011

POWERED BY
SEDONA
FRAMEWORK™

Powered by
niagara ^{AX}
FRAMEWORK®

Sedona Framework Sox Tools Guide

Confidentiality Notice

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information, and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark Notice

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and Visio are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are registered trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, Niagara^{AX} Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlace^{AX}, and ^{AX}Supervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that is known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and Patent Notice

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.

Copyright © 2011 Tridium, Inc.

All rights reserved. The product(s) described herein may be covered by one or more U.S or foreign patents of Tridium.

CONTENTS

Preface	iii
About this document	iii
Sox Tools terms	iii
Document Change Log	iv
About The Provisioning Tools View And Schema	1-1
Sox Tools	1-1
Schema	1-2
About The App Manager	2-1
<i>Copy the app from a device</i>	<i>2-1</i>
<i>Copy an app to a device</i>	<i>2-2</i>
About The Backup/Restore Tool	3-1
<i>Create a backup</i>	<i>3-1</i>
<i>Restore a backup</i>	<i>3-2</i>
About The Kit Manager	4-1
<i>Make kit selection changes</i>	<i>4-3</i>
FAQs and Troubleshooting	A-1
Frequently asked questions	A-1
Adding a kit operation fails	A-1
“No tools available” message displays	A-1
<i>Install a PAR file</i>	<i>A-2</i>
“Provisioning Failed” result during backup	A-2

PREFACE

Preface

This documents the Sedona Framework Sox Tools available when making a Sox connection to a Sedona Framework device, when using either NiagaraAX Workbench 3.6 or later with Sedona Framework TXS 1.1, or when using Sedona Framework Workbench 1.1. Once Sox-connected to the device, both Workbench products provide the identical Sox Tools for provisioning the Sedona Framework device.

This preface has the following sections:

- [About this document](#)
- [Sox Tools terms](#)
- [Document Change Log](#)

About this document

This document describes the Sox provisioning tools available when connected to a Sedona Framework device and has the following main sections:

- [“About The Provisioning Tools View And Schema”](#)
Explains the Provisioning Tools view and describes the provisioning tools and schema.
- [“About The App Manager”](#)
Explains the App Manager tool and describes the procedures for copying an app from or to a Sedona Framework device.
- [“About The Backup/Restore Tool”](#)
Explains the Backup/Restore tool and describes the procedures for backing-up and restoring Sedona Framework device components.
- [“About The Kit Manager”](#)
Explains the Kit Manager tool and describes the procedure for making kit selection changes on a Sedona Framework device.
- [“FAQs and Troubleshooting”](#)
Provides troubleshooting tips.

Sox Tools terms

The following is a list of terms and abbreviations used in this document when describing Sedona Framework Sox Tools and Sedona Framework devices. Complete Sedona Framework documentation is available at SedonaDev.org. For general NiagaraAX terms, see the Glossary in the *Niagara AX User Guide*.

app The app in a Sedona Framework device is its collection of Sedona Framework components, including links between them, plus configuration properties.

bundle Sedona Framework TXS software is distributed in a “bundle”. A bundle is a special image (set of files) that is available from Niagara-Central (www.nigara-central.com). You use the Sedona Installer tool in Workbench to install a bundle. For complete details, refer to the *NiagaraAX Sedona Installer Guide*.

DASP For Datagram Authenticated Session Protocol. This is the low-level, secure session-based protocol that Sox utilizes.

kit Sedona Framework kits are the basic unit of modularity of Sedona Framework software, encapsulating code, types, and metadata. A kit is analogous to a module on a NiagaraAX platform. The app in a device instantiates components and services contained in its installed kits. You must have the appropriate

kits available on your Workbench to change a device's "core" software. Sedona Framework Sox Tools in Workbench include a "Kit Manager" view to manage kits on a Sedona Framework device.

manifest Each kit has a corresponding manifest, with all metadata needed by tools like Workbench to connect via Sox to a Sedona Framework device, and for a station to support Sedona Framework proxy points in that device. Manifest files are compact XML files, named using a kitName-checksum convention similar to kit files, but with an .xml extension. Use the Manifest Manager view of either Sedona Framework network type to manage kit manifests on the NiagaraAX host (JACE or Supervisor). Use the Sedona Manifest Manager view in Workbench tools to manage manifests on your Workbench host. For more details see the *Sedona Manifest Manager* document.

PAR For Platform Archive, or PAR file. This is a zip file with a .par extension that provides a way of organizing various files and metadata about a platform into a single entity. The contents of the platform archive is an .xml file that describes the platform.

SAB For Sedona App Binary. This is a compact binary representation of a Sedona Framework app suitable for storage and execution on a Sedona Framework device. One of two file formats (.sab) for a Sedona Framework app.

SAX For Sedona App XML. This is a simple XML representation of a Sedona Framework app that is easily generated and consumed by Sedona Framework software tools. One of two file formats (.sax) for a Sedona Framework app.

schema The specific collection of Sedona Framework kits installed in a device (each at some particular checksum revision) is called a schema. You see the schema table when you click on the Tools node when Sox-connected to a device.

scode A compiled set of kits in a Sedona Framework code image file with an .scode extension, the scode image is executed by the SVM in the device. The Kit Manager tool is used to modify kits.scode on a Sedona Framework device.

SEDONA HOME The term SEDONA HOME is a system "variable" for the installation-specific directory of your Workbench computer's "root" folder for Sedona Framework files used in Sox connections from Workbench. To see this location, search for "sedona.home" in the !/lib/system.properties file. When using the Sedona Installer to install a bundle, the SEDONA HOME system variable is set correctly for you in the system.properties file.

Sox Sox is the standard protocol used to communicate with Sedona Framework devices. It runs over UDP via the lower-level DASP protocol. Workbench always uses Sox to connect to (open) a Sedona Framework device. A Niagara station also uses Sox to discover and (typically) read and write to Sedona Framework proxy points.

SVM The SVM (Sedona Virtual Machine or VM) executes the compiled kits scode image in a device along with its app.

Document Change Log

Updates (changes/additions) to this *Niagara^{AX} Sedona Framework Sox Tools Guide* document are listed below.

- Publication: November 7, 2011
Initial document.

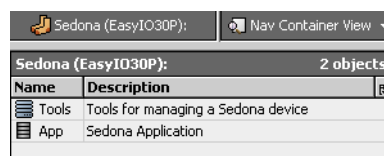
CHAPTER 1

About The Provisioning Tools View And Schema

The initial Sox connection shows the Tools container and App component in a Nav Container view, as shown below in [Figure 1-1](#). Double-click the Tools row to access the Provisioning Tools.

Note: The Provisioning Tools can also be accessed by expanding the Tools container in the Nav Tree or clicking the Nav Container view drop-down list in the upper right corner.

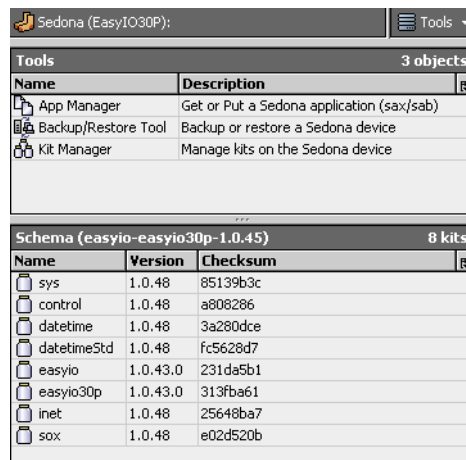
Figure 1-1 Tools view



Sedona (EasyIO30P): 2 objects	
Name	Description
Tools	Tools for managing a Sedona device
App	Sedona Application

The Provisioning Tools view is the default view of the Tools container. This view consists of the Tools and Schema tables shown in [Figure 1-2](#). Use this view for all Sedona Framework platform provisioning of the device, including all management of apps and kits, as well as performing backup and restore operations.

Figure 1-2 Provisioning Tools view



Tools 3 objects	
Name	Description
App Manager	Get or Put a Sedona application (sax/sab)
Backup/Restore Tool	Backup or restore a Sedona device
Kit Manager	Manage kits on the Sedona device

Schema (easyio-easyio30p-1.0.45) 8 kits		
Name	Version	Checksum
sys	1.0.48	85139b3c
control	1.0.48	a808286
datetime	1.0.48	3a280dce
datetimeStd	1.0.48	fc5628d7
easyio	1.0.43.0	231da5b1
easyio30p	1.0.43.0	313fba61
inet	1.0.48	25648ba7
sox	1.0.48	e02d520b

See the following sections for more details:

- [Sox Tools](#)
- [Schema](#)

Sox Tools

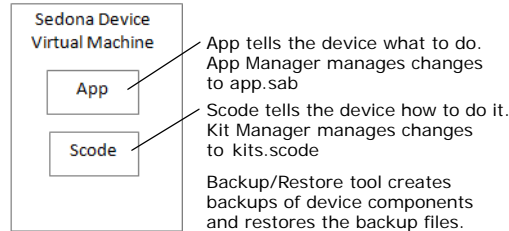
The three tools available in the Provisioning Tools view are:

- **App Manager**
Manages applications on a Sedona Framework device. App Manager uses the Get and Put functions to copy apps from or to a device.

- **Backup/Restore Tool**
Creates backup files for Sedona Framework device components as zip files and restores the backups. These backup files can contain the Sedona virtual machine (SVM), the app.sab and kits.scode, as well as the kits and manifests.
- **Kit Manager**
Manages the kits.scode on a Sedona Framework device. This tool is used to add and remove kits, and to change the kit versions running on a device.

The diagram in [Figure 1-3](#) below shows the relationship of the provisioning tools to the components of the Sedona virtual machine.

Figure 1-3 Relationship of Provisioning Tools to SVM components



The provisioning tools are wizard-type tools which simplify the provisioning process by presenting you with a sequence of dialog boxes that lead you through a series of steps. Use the controls (a set of buttons positioned at the bottom of the window) shown in [Figure 1-4](#) to navigate through the individual steps of a procedure.

Figure 1-4 Provisioning Tools controls



The buttons function as follows:

- The **Next** button advances to the next step in a procedure.
- The **Back** button returns to the previous step in a procedure.
- The **Finish** button completes a procedure.
- The **Cancel** button aborts a procedure.

Schema

The schema is the collection of “kits” used by the app currently running on a device. The schema can be viewed in the Schema table located in the lower portion of the Provisioning Tools view, as shown in [Figure 1-2](#). Each row of the Kits table represents a kit. A kit contains some number of named types, where each type declares zero or more slots. For any given version of a kit, there is a fixed list of types and their declared slots. The checksum for each kit is generated when the Sedona Framework compiler creates a kit zip file. The specific list of kit parts (kits at a specific checksum revision) is called a schema.

CHAPTER 2

About The App Manager

The App Manager tool is used to copy an app from or to a Sedona Framework device using the Get and Put functions. When launched, App Manager displays the Get/Put Application wizard which leads you through the steps of each procedure.

Some reasons for using the Get option are:

- **To modify the app.**
You may want to make app modifications offline.
- **To save the app as a backup.**
Although, it may be more useful to make a complete backup using the Backup/Restore Tool.
- **To distribute the app.**
You may want to distribute the app to another device.

The reason for using the Put option is to replace the app that is running on a device.

The following procedures are described below:

- [Copy the app from a device](#)
- [Copy an app to a device](#)

Copy the app from a device

This procedure copies the app from a Sedona Framework device to the Workbench using the Get option.

- Step 1 In the Nav Container view, double-click the Tools row to display the Provisioning Tools view.
- Step 2 Double-click **App Manager** to launch the Get/Put Application wizard.
- Step 3 Click the **Get** option. You are prompted for the location and file name to use when saving the app. By default, the \niagara-3.6.xx\backups\sedona\apps folder is used, with a default app file name of appYYYYMMDD.sax, where YYYY is year, MM is month, DD is day, as shown in [Figure 2-1](#).

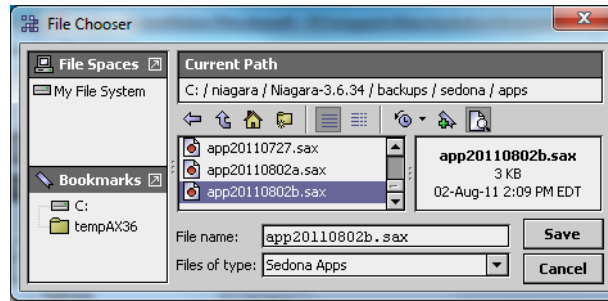
Figure 2-1 Get dialog box



Note: By default, the option to “Save kit checksums in sax file” is checked. If checksums are saved, then the sax file will only run with those exact versions of kits. If checksums are not saved, the sax file can be installed onto a device with the required kits without regard to their checksum. During development, if you are regularly making changes to the kits, then you may want to uncheck this option.

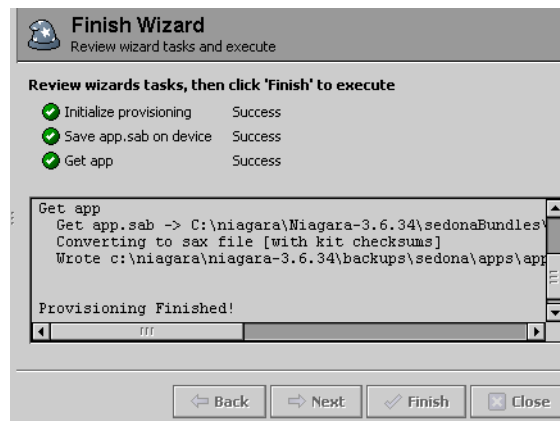
As needed, click the **Browse** button for the standard **File Chooser** dialog box to change location and/or filename, as shown in [Figure 2-2](#). Then click the **Save** button to close the dialog box.

Figure 2-2 File Chooser dialog box during Get function



- Step 4 Click the **Next** button. The tasks to be performed are listed in the upper portion of the screen.
- Step 5 Click the **Finish** button to execute the tasks. As each task is completed, the status (Success or Failed) is displayed to the right of the task. Also, a log can be viewed in the pane. On completion you are notified that provisioning has finished, as shown in [Figure 2-3](#).

Figure 2-3 Get function completed successfully.



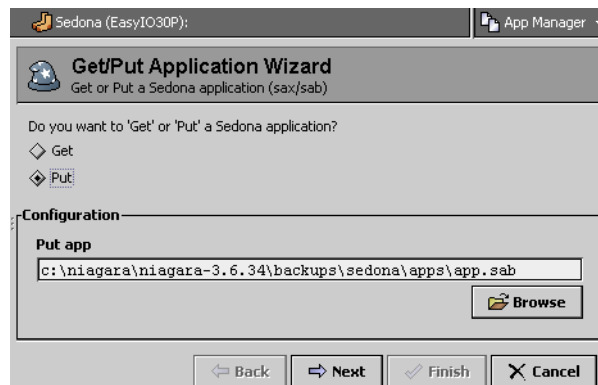
- Step 6 Click the **Close** button to return to the Tools view.

Copy an app to a device

This procedure copies an app to a Sedona Framework device from the Workbench using the Put option.

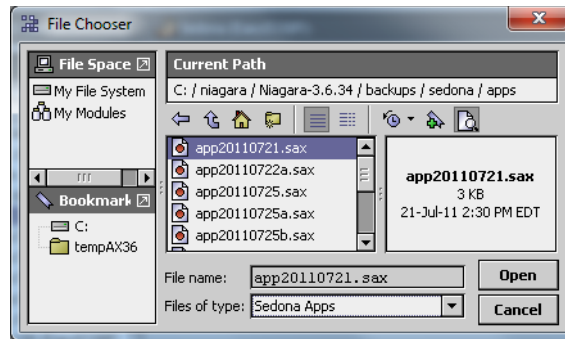
- Step 1 In the Nav Container view, double-click the Tools row to display the Provisioning Tools view.
- Step 2 Double-click **App Manager** to launch the Get/Put Application wizard.
- Step 3 Click the **Put** option. You are prompted for the location and file name of the app to be copied to the device. By default, the \niagara-3.6.xx\backups\sedona\apps\app.sab location and file name are used, as shown in [Figure 2-4](#).

Figure 2-4 Put dialog box



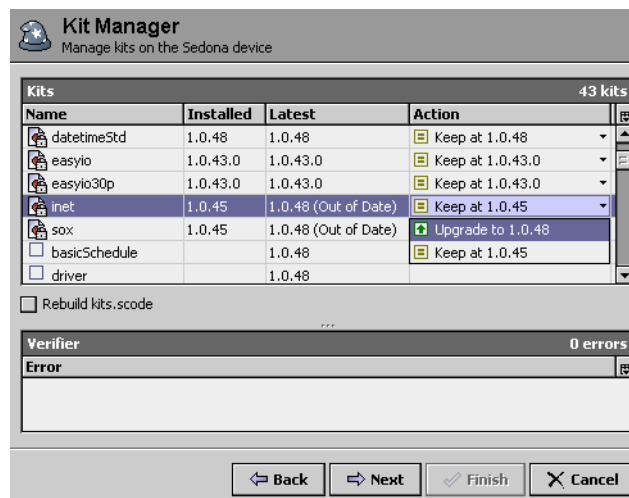
- Step 4 Click the **Browse** button for the standard **File Chooser** dialog box to find and select the app. By default, the **File Chooser** looks for a .sax file in the /Niagara-3.6.xx/backups/sedona/apps folder, as shown in [Figure 2-5](#). Select the app and click the **Open** button to close the dialog box.

Figure 2-5 File Chooser dialog box during Put function



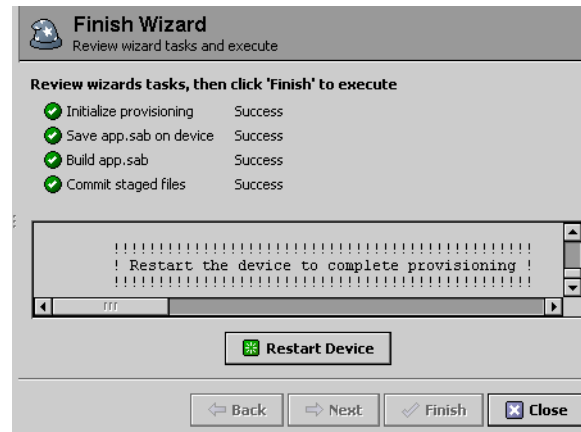
- Step 5 Click the **Next** button. The Kit Manager launches, see [“About The Kit Manager”](#). The Kit Manager ensures that the appropriate kits will be deployed on the device in order to support the app. If needed, change any installed kit’s version level by clicking on the individual Action value to reveal the **Action** drop-down list, as shown in [Figure 2-6](#).

Figure 2-6 Action on each installed kit lets you keep at current kit version, upgrade or downgrade



- Step 6 If the Verifier table is free of errors, and kits are at the desired version levels, click the **Next** button. The tasks to be performed are listed in the upper portion of the screen.
- Step 7 Click the **Finish** button to execute the tasks. As each task is completed, the status (Success or Failed) is displayed to the right of the task. Also, a log can be viewed in the pane. On completion you are prompted to restart the device.

Figure 2-7 Put function completed successfully



Step 8 Click the **Restart Device** button to restart the device.

Note: Restarting the device will disconnect your connection. To re-open your Sox connection, double-click the dimmed Sedona node in the Nav Tree, and enter the log-in credential.

CHAPTER 3

About The Backup/Restore Tool

The Backup/Restore Tool is used to backup Sedona Framework device components in a zip file that is saved to the Workbench, and to restore a backup file to a Sedona Framework device. The backup file can contain the Sedona virtual machine (SVM), the app.sab and kits.scode, as well as all of the kits and manifests that your app requires to run if deployed on another device. In this sense, the backup tool provides a true “backup”. Once you have created a backup of a device, if the device fails you can quickly get operations running again by restoring the backup to a replacement device.

The three backup options that are available:

- **Backup Sedona VM**
This option backs-up the virtual machine.
Note: This option is device-specific. If not supported by your device, selecting the option will cause the backup operation to fail. Refer to your device documentation for information on backing up the virtual machine.
- **Backup app.sab and kits.scode**
By default, this option is selected. It provides a minimal backup that contains only the app.sab and kits.scode.
- **Backup kits and manifests the device depends on**
This option includes all additional files needed for the app to run on another device.
Note: This option requires the backup of the app.sab and kits.scode as well.

To restore a backup, simply connect to the device and provide the location and file name of the zip file backup to be restored.

The following procedures are described below:

- [Create a backup](#)
- [Restore a backup](#)

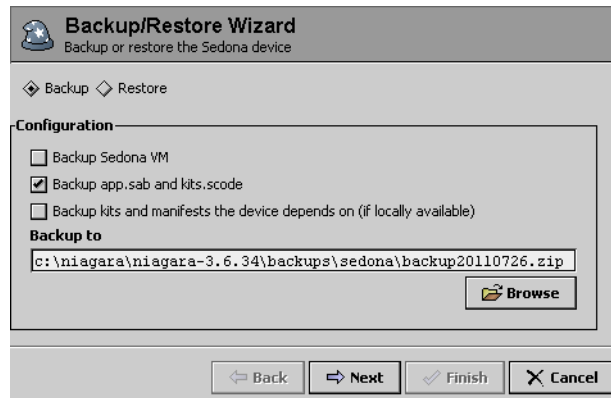
Create a backup

This procedure creates a zip file backup of selected components on a Sedona Framework device and saves the zip file to the Workbench.

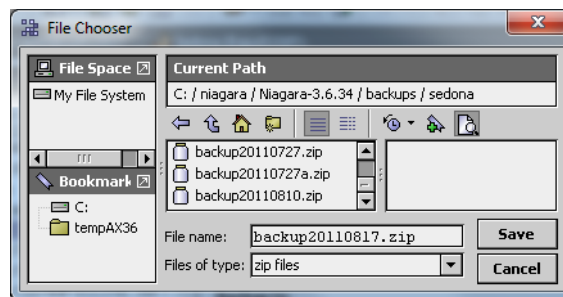
- Step 1 In the Nav Container view, double-click the Tools row to display the Provisioning Tools view.
- Step 2 Double-click the **Backup/Restore Tool** to launch the Backup/Restore Wizard. By default, the **Backup** option is selected.
- Step 3 Click configuration check boxes to select the device components to be backed-up to your local file system and provide the location and file name for the backup zip file.

Note: The first option, “Backup Sedona VM”, is device-specific. If not supported by your device, selecting it will cause the backup operation to fail. Refer to your device documentation for information on creating a backup of the virtual machine.

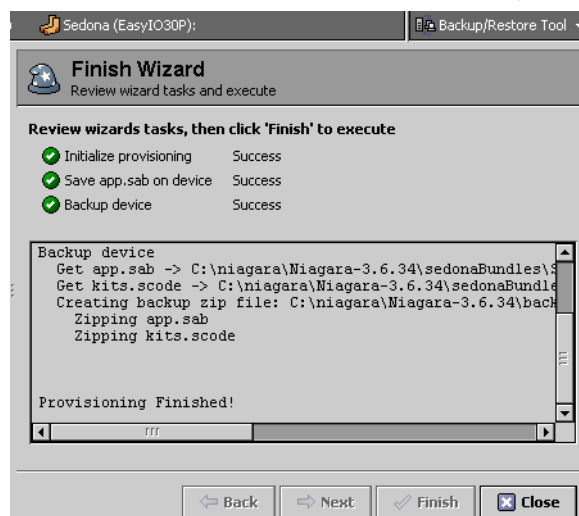
By default, the option to “Backup app.sab and kits.scode” is selected, and the \niagara-3.6.xx\backups\sedona\ folder is shown, with a default backup file name of backupYYYYMMDD.zip, where YYYY is year, MM is month, DD is day, as shown in [Figure 3-1](#).

Figure 3-1 Backup view with default selection

As needed, use the **Browse** button for the standard **File Chooser** dialog box to change location and/or file name and click the **Save** button to close the **File Chooser** dialog box, as shown in [Figure 3-2](#).

Figure 3-2 File Chooser dialog box during Backup function.

- Step 4 Click the **Next** button. The tasks to be performed will be listed in the upper portion of the screen.
- Step 5 Click the **Finish** button to execute the tasks. As each task is completed, the status (Success or Failed) is displayed to the right of the task. Also, a log can be viewed in the pane. On completion you are notified that provisioning has finished, as shown in [Figure 3-3](#).

Figure 3-3 Backup function completed successfully

- Step 6 Click the **Close** button to exit the Backup/Restore Wizard.

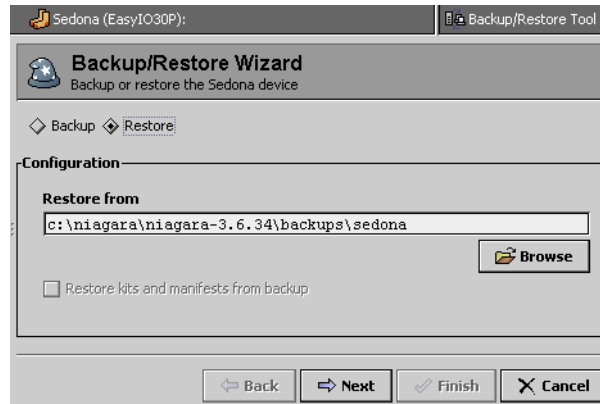
Restore a backup

This procedure restores a zip file backup by saving the contents of the selected zip file to a Sedona Framework device.

- Step 1 In the Nav Container view, double-click the Tools row to display the Provisioning Tools view.
- Step 2 Double-click the **Backup/Restore Tool** to launch the Backup/Restore wizard.

- Step 3 Click the **Restore** option. You are prompted to provide the location and file name for the zip file backup to install on the device, as shown in [Figure 3-4](#).

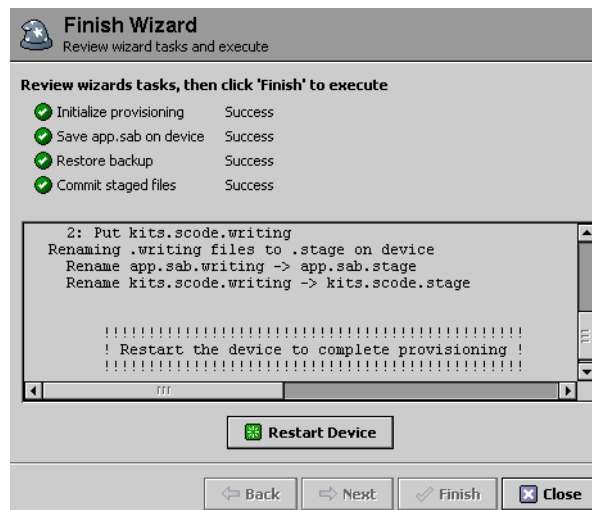
Figure 3-4 Restore dialog box with default selection



If the backup file includes the kits and manifests, you will have the option to restore those or not. Click the **Restore kits and manifests from backup** check box if you want to restore those. They will be installed, overwriting the kits and manifests currently on the Sedona Framework device.

- Step 4 Click the **Browse** button for the standard **File Chooser** dialog box to change location and/or file name and click the **Open** button to close the **File Chooser** dialog box.
- Step 5 Click the **Next** button. The tasks to be performed will be listed in the upper portion of the screen.
- Step 6 Click the **Finish** button to execute the tasks. As each task is completed, the status (Success or Failed) is displayed to the right of the task. Also, a log can be viewed in the pane. When the restore operation finishes, you are prompted to restart the device to complete provisioning, as shown in [Figure 3-5](#).

Figure 3-5 Restore function completed successfully



Note: Restoring a backup requires restarting the device, which disconnects your connection. To re-open your Sox connection, double-click the dimmed Sedona node in the Nav Tree, and enter the log-in credentials.

- Step 7 Click the **Restart Device** button to exit and restart the device.

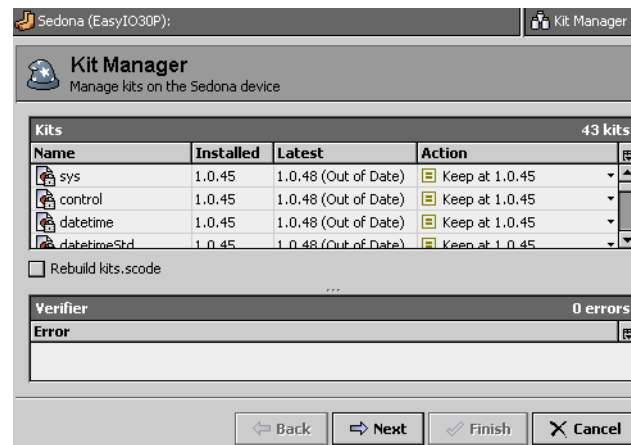
CHAPTER 4

About The Kit Manager

The Kit Manager tool is used to make changes to the kits.score on a Sedona Framework device. Using the Kit Manager, you can add and remove kits, as well as change version levels of kits deployed in the score running on the device. Also, you have the option to force the kits.score to be rebuilt.

Note: *The appropriate kits must be available on your Workbench in order to change the kits on a device using the Kit Manager. All of the open source kits and Tridium specific kits are provided in the TXS bundle installation. Third party vendors provide their own kits and instructions for downloading and installing them.* When launched, Kit Manager analyzes the kits installed on a device. A progress bar displays in the view during this analysis. Afterwards, the Kit Manager view displays featuring the Kits table and Verifier table, as shown in [Figure 4-1](#).

Figure 4-1 Kit Manager view



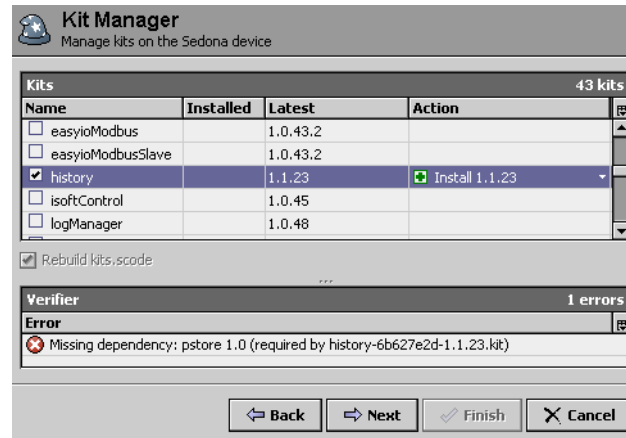
The Kits table shows the schema or collection of “kits” used by the app currently running on the device. Each row in the Kits table represents a kit. Also shown for each kit, is the specific version installed on the device, the latest version downloaded to the Workbench, as well as the individual actions available for the kit.

In the Kits table, a kit icon with a padlock (🔒) shown to the left of the kit name indicates that the kit is required by the app running on the device and may not be deleted. An unchecked check box to the left of a kit name indicates that the kit is compatible and may be installed. A checked box indicates a kit that is installed, but not used by the device. The kit may be uninstalled, if desired. Kits that are not supported by the device appear dimmed on a gray background. The Action column describes those kits as “Not supported”.

Between the two tables, is a check box for the option to **Rebuild kits.score**. The kits.score must be rebuilt whenever a change affects the schema. This ensures such changes are included in the recompiled binary .score image. When you make any kit selection changes in the Kit Manager, it automatically selects and dims this option. The only time you would select this option is when you want to force the score to be rebuilt (even if no change was made to kit selections). This option is most useful in a development environment where the actual code in a kit might change while the kit version and checksum do not. In this case, no change is required in the Kit Manager but the score must be regenerated in order to deploy the new code on a device.

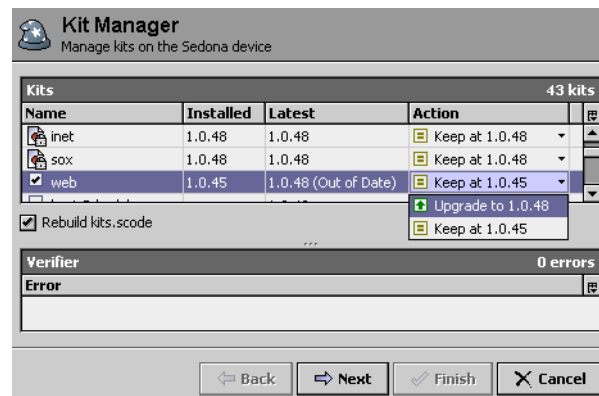
The Verifier table shows kit dependency errors when they occur. If a selected kit requires the presence of an uninstalled kit in order to run on the device, the missing dependency will be listed in the Verifier table, as shown in [Figure 4-2](#). When a kit dependency error occurs, the Kit Manager will not automatically select a missing kit (even if it is present on the Workbench and available for installation) since doing so could cause unexpected results. The missing kit, if not present in the Workbench, must be obtained from the vendor.

Figure 4-2 Verifier table shows a missing dependency (pstore 1.0) for the history kit.



The actions that are available for each installed kit are listed in the **Action** drop-down list which you access by clicking a kit's Action value, as shown in [Figure 4-3](#).

Figure 4-3 Kit Manager dialog box



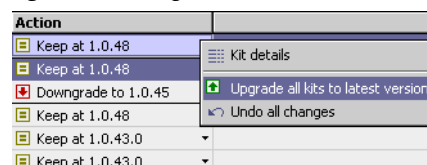
The **Action** drop-down list options can be one or more of the following:

- **Keep at** [the current version]
- **Upgrade to** [a newer version]
- **Downgrade to** [an older version]

The newer/older kit version levels shown in the **Action** drop-down menu are available in the Workbench. They are stored in the `SEDONAHOME\kits` directory.

If you have older kits in the current app, you may want to upgrade all kits to the newer versions available in your Workbench. You can easily do this by right-clicking within the Kits table to display the drop-down list shown in [Figure 4-4](#), and then select the option to **Upgrade all kits to latest version**.

Figure 4-4 Right-click in the Kits table to display the Kit details drop-down list



The **Kits table** drop-down list options function as follows:

- **Kit details**

This option opens a pop-up window listing details for the selected kit. The details include kit name, current version with a drop-down list for other versions (if available), kit checksum as well as listing any kit dependencies.

- **Upgrade all kits to latest version**
This option changes the Action value for all installed kits to **Upgrade to** (latest version)
- **Undo all changes**
This option, if selected before clicking the **Next** button, reverses the kit version changes.

Make kit selection changes

This procedure makes kit selection changes (add/remove kits or change kit version levels) to kits deployed on a Sedona Framework device.

- Step 1 In the Nav Container view, double-click the Tools row to display the Provisioning Tools view.
- Step 2 Double-click Kit Manager. After analyzing the kits currently installed on the device, the Kit Manager view is displayed.
- Step 3 In the Kits table, make selections as needed to add or remove kits, and to change kit versions.
- Add a kit by clicking on the kit's unchecked check box.
 - Remove an installed kit by clicking the kit's checked check box to clear it.
 - Change a kit version by clicking on the kit's Action value.
- Note:** *The option to "Rebuild kits.scode" is selected automatically whenever you change kit selections and/or change kit versions since the schema will need to be updated.*
- Step 4 Click the **Next** button. The tasks to be performed are listed in the upper portion of the view.
- Step 5 Click the **Finish** button to execute the tasks. As each task is completed, the status (Success or Failed) is displayed to the right of the task. Also, a log can be viewed in the pane. On successful completion you are prompted to restart the device.
- Step 6 Click the **Restart Device** button.
- Note:** *Restarting the device will disconnect your connection. To re-open your Sox connection, double-click the dimmed Sedona node in the Nav Tree, and enter the log-in credentials.*

APPENDIX A

FAQs and Troubleshooting

Frequently asked questions about the provisioning tools and troubleshooting tips for problems that you may experience while provisioning a Sedona Framework device are covered in this section.

The following topics are discussed below:

- [Frequently asked questions](#)
- [Adding a kit operation fails](#)
- [“No tools available” message displays](#)
- [“Provisioning Failed” result during backup](#)

Problems not listed here may be network issues. For information on Sedona Framework networks, refer to the *Niagara AX Sedona Framework Networks Guide*.

Frequently asked questions

Below are a few frequently asked questions (FAQs) about the Sox Tools.

Q: Is there any difference between the Sox tools that are available in Sedona Framework TXS-enabled Niagara Workbench and those available in Sedona Workbench?

A: There is no difference. The tools are the same in both versions of Workbench.

Q: What is a PAR file?

A: The Platform Archive (PAR) file is a zip file with a .par extension. The PAR file provides a way of organizing various files and metadata about a platform into a single entity. Basically, it describes the platform. For more information on Platform Archive, see SedonaDev.org.

Q: Is there a relationship between the schema of a Sedona Framework device and its platform archive, and if so, what?

A: No, the schema is completely independent of the platform archive. For more information on schema and platform archive, refer to the [“Sox Tools terms”](#) in the Preface section.

Q: How can I avoid the situation where a kit fails to install on a Sedona Framework device due to insufficient space?

A: Currently, there is no way to avoid this. The maximum file size is device-dependent. We do not have a way to know a device's maximum file size.

Adding a kit operation fails

This can occur if the kit you are attempting to add exceeds the maximum file size for the device. Refer to the vendor's documentation for information on the maximum file size for the device.

“No tools available” message displays

The “No tools available” message displays when attempting to open the Provisioning Tools view after connecting to a Sedona Framework device. This occurs when the Platform Archive (PAR) file for the connected Sedona Framework device is missing from the platform database. The PAR file is required by the Tools. The PAR file describes a platform, similar to the way a manifest describes a kit. Vendors should include a PAR file as part of their build. For more information on PAR files, see [SedonaDev.org \(http://sedonadev.org/doc/par.html\)](http://sedonadev.org/doc/par.html).

Install a PAR file

This troubleshooting procedure corrects the missing PAR file problem by installing the PAR file in the proper location in the platform database. This procedure assumes you have obtained the missing PAR file from the vendor or other source.

- Step 1 Open the platformManifest.xml file (located in the root of the PAR file) and find the platform id attribute. The platform id describes a directory structure that shows where you must locate the PAR file.
Example: The platform id will look similar to this: acme-basicPlatform-win32-1.0.38, where every '-' represents a directory separator.
- Step 2 Using the platform id as a guide, build that directory structure in `SEDONA_HOME/platforms/db/`.

Figure A-1 Directory structure for the platform id example shown in Step 1.

```
sedona_home/platforms/db/
+- acme/
| +- basicPlatform/
| | +- win32/
| | | +- 1.0.38/
```

- Step 3 In the last subdirectory, create a directory named '.par'.
- Step 4 Copy your PAR file to the .par directory and unzip it there.

Figure A-2 Directory structure after unzipping the PAR file (based on platform id example in Step 1).

```
sedona_home/platforms/db/
+- acme/
| +- basicPlatform/
| | +- win32/
| | | +- 1.0.38/
| | | | +- .par/
| | | | | +- platformManifest.xml
| | | | | +- svm/
| | | | | | +- <svm binary>
```

- Step 5 Re-connect to the device and Nav Container view, double-click the Tools row to open the Provisioning Tools view.

“Provisioning Failed” result during backup

The complete error message states: “sedona.sox.SoxException: Request failed: cannot open” followed by “Provisioning Failed”. This error will occur when attempting to backup the Sedona VM on a device that does not support it. The option to backup the Sedona VM is device-specific. Certain devices support it while others do not. Refer to the vendor’s documentation for information on making a backup of the virtual machine.