TIBCO iWay® Service Manager

Non-blocking AS2 (NAS2) Getting Started Guide

Version 7.0.x and Higher
March 2021
DN3502300.0321

Contents

Chapter 1

Introducing Non-blocking AS2 (NAS2)

This section provides an overview of the providers and namespaces that are used with Non-blocking AS2 (NAS2).

In this chapter:

NAS2 Overview

The NAS2 adapter is a non-blocking AS2 with improved performance, connection management, and various other security features.

The NAS2 adapter provides extensive flexibility by exposing an array of parameters that can be configured for security providers, Message Disposition Notification (MDN) handling, Certificate Revocation List (CRL) checking, and so on.

Features

This section describes the features that have been added as part of the improvement to the NAS2 adapter.

- **LDAP Certificate Support.** Retrieval of partner certificates from the LDAP system as part of the certificate store configuration to complete the signature chain validation.
- **Signer Certificate Chain.** Option to not include the signer certificate when sending an AS2 message or replying with an MDN. This allows you to minimize the message size for enhanced performance.
- **Certificate Revocation List Checking Option.** Allows the configuration of NAS2 to validate if the message being processed is signed using a revoked certificate. If the option for CRL checking is selected, it will require a configured certificate store on the NAS2 component which can point either to a list of named keystore providers, directory CertStore providers, and directory Providers (LDAP) where the revoked certificates are located.

- **Key Alias Selection.** On the S/MIME and SSL components, new parameters are exposed, which allow you to specify the key alias with the keystore and truststore. This allows you to pick which key to use for various security operations such as signatures, decryptions, and so on.
- **Persistent Connection Support.** The NAS2 adapter supports persistent connections, which allows improved connection handling and management.
- **Ordering of Signature and Compression.** A feature to allow the selection of compression and signature ordering is available. Now you can configure if the message should be signed and then compressed or compressed then signed.
- **Delayed MDN.** The NAS2 adapter also supports the feature which is not typical to the standard AS2 processing, but allows a great degree of flexibility when it comes to MDN processing. When a message is received on the NAS2 listener, you may configure the MDN to be delayed until the business processing of the message is completed. If the Delayed MDN option is selected, it is your responsibility to invoke the corresponding MDN send service as part of the business processing that will send the MDN as requested by the originator of the message.
- **Safe Store for Messages.** The Safe Store option on the NAS2 component will safe store the message before performing any further processing to the message. This will prevent any message loss. After the message has been processed, it will be removed from the safe store. In the event that the system goes down, all the messages in the safe store will be processed after the system is back on line.
- **Large File Limit.** The NAS2 adapter contains various internal improvements to handle large file sizes. An option exposed on the NAS2 inbound processing that allows you to limit the message size accepted by the NAS2 adapter.

NAS2 Providers Overview

A provider is a centrally configured resource that supplies services to run time components in the server. For example, a keystore provider centralizes the definition of one security keystore, including its type, file location, and password.

A provider is referenced by name in components that require its services, which allows re-usability and less complexity on the component's configuration.

One provider can refer to another provider. For example, the SSL provider requires keystore and truststore providers that it references by name.

The following is a list of providers and their functions.

- **Security Providers**
  - **Keystores.** Standard repositories of security certificates that are used in encryption and digital signature operations.
  - **SSL Context.** Defines the parameters used for transport layer security. Incorporates required keystore providers.
  - **Directory Certstore.** Defines directories from which certificates and CRLs can be loaded into a certificate store.
- **Directory Provider.** Directories house information organized by keys and context. The most commonly used directories are accessed through LDAP.
- **XML Namespace Map Provider.** Defines a set of xmlns prefixes and URIs for XML namespaces. Used by components such as XMLDSig Services.
- **Pooling Provider.** The HTTP Client provider allows HTTP connections to be shared among iSM components. An instance of the HTTP Client Provider represents a pool of connections.

The following is a list of the security providers and their functions:

- **Keystore.** Is a database of key material used for authentication and data integrity check. Some keystores can contain both encryption keys and security certificates. Formally, however, a keystore holds the private key for one or more PKI key pairs.
- **Truststore.** Is a database of key material same as keystore. It holds the public certificates of trusted partners. Although it is possible to share a single file with the keystore, formally a truststore and a keystore are separate entities.
- **Certstore.** Is a database of public key certificates and Certificate Revocation Lists used for CRL checking.

NAS2 Namespaces Overview

Special register names can be preceded with a namespace prefix. The namespace is the first part of the name before the first decimal. For example, register xyz is in the default namespace, while register abc.xyz is in the abc namespace.

The NHTTP/NAS2 components support register namespaces through the configurable parameters for request and response headers.

The SREG namespace service (com.ibi.agents.XDSREGNamespaceAgent), is available to perform the following operations on registers in namespaces:

- **Copy.** Duplicates registers from a source namespace to a destination namespace. After a copy operation is performed, the registers are available in both namespaces.
- **Move.** Moves registers from one namespace to another.
- **Delete.** Deletes all registers in the namespace.
- **Exist.** If any registers exist in the namespace, the flow is passed down the success edge, otherwise it is passed down the notfound edge.

Chapter 2

Configuring NAS2 Providers

This section describes how to create and use providers for NAS2.

In this chapter:

NAS2 Prerequisites

Before using NAS2, ensure that the prerequisites described in this section are met.

Java Development Kit (JDK)

You must have JDK version 1.6.0_21 or higher installed.

Unlimited Strength Java Cryptography Extension (JCE) Policy Files

You must download the unlimited strength JCE policy files for JDK 6:

- local_policy.jar
- US_export_policy.jar

These files can be downloaded from *http://java.sun.com*.

Once the files are downloaded, copy them to \jre\lib\security and overwrite the existing versions of these .jar files.

Note: If you are using Windows, the JDK installation will install a JRE and a JDK environment in two separate locations. For example:

```
C:\Program Files\Java\jre6\lib\security

C:\Program Files\Java\jdk1.6.0_21\jre\lib\security
```

You must install the unlimited strength JCE policy files in both environments.

Bouncy Castle as a Security Provider

To install Bouncy Castle as a security provider, you must add it as an entry to the java.security file, which is located in the following directory for the JDK/JRE that you are using:

```
$JAVA_HOME/jre/lib/security/java.security
```

Look for a list of lines starting with security.provider.*X* where *X* is a number.

Add the following line at the bottom of the list:

```
security.provider.N=org.bouncycastle.jce.provider.BouncyCastleProvider
```

where:

N

Is one more than the previous number in the list.

For example:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
security.provider.4=com.sun.crypto.provider.SunJCE
security.provider.5=sun.security.jgss.SunProvider
security.provider.6=com.sun.security.sasl.Provider
security.provider.7=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.8=sun.security.smartcardio.SunPCSC
security.provider.9=sun.security.mscapi.SunMSCAPI

security.provider.10=org.bouncycastle.jce.provider.BouncyCastleProvider
```

It is possible to add the provider higher up in the list. If you do this, iWay recommends that you do not add it earlier than position 2 as there are occasionally internal dependencies on the provider at position 1 that may cause some operations by your Java environment to result in errors.

Importing Packages

The NAS2_Provider-package.zip file that is attached to this document contains the following pre-configured security providers:

- iWaySMIMEKeyStore
- iWaySMIMETrustStore
- PartnerSMIMEKeyStore
- PartnerSMIMETrustStore

This package also contains a pre-configured HTTP connection pool provider (httpcon). This provider is used within the NAS2 emit service to manage connection pools and provide reusability of the configuration.

This section describes how to import the NAS2_Provider-package.zip file as a package using the iWay Service Manager Administration Console.

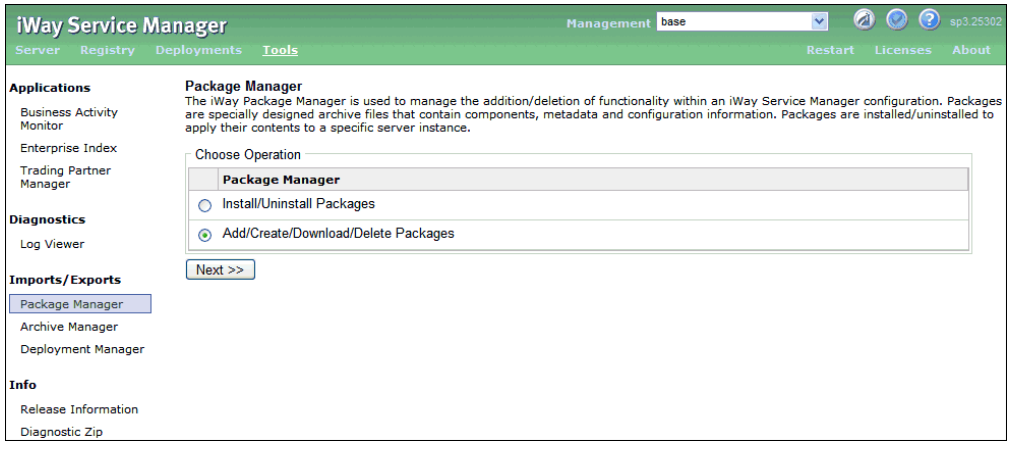
Procedure: How to Import a Package

To import a package:

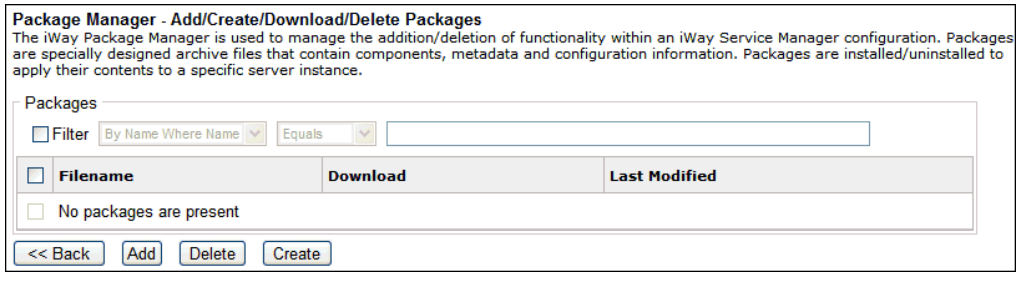
1. Navigate to *Tools, Package Manager* in the iSM console.

   The Package Manager screen is displayed.

2. Select the second option, Add/Create/Download/Delete Packages, and then click *Next*, as shown in the following image.



   The Add/Create/Download/Delete Packages pane is displayed.

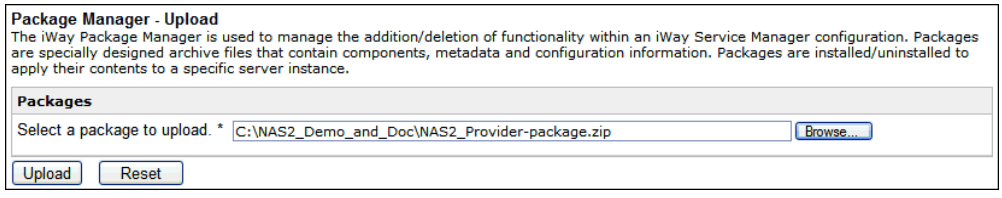


3. Click *Add* and browse to the NAS2_Provider-package.zip file on your file system.

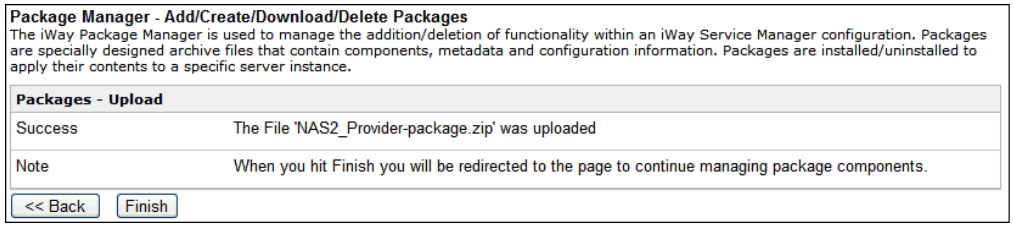
   Note: For your convenience and for reference purposes, a copy of the NAS2_Provider-package.zip file is attached to this document. The NAS2_Provider-package.zip file contains the four security providers and an HTTP connection pool provider that are used and referenced in this NAS2 configuration.

For PDF-compatibility purposes, the file extension of the NAS2_Provider-package.zip file is temporarily renamed to **.zap**. After saving this file to your file system, you must rename this extension back to **.zip** before it can be imported to iWay Service Manager.

4. Click *Upload*.

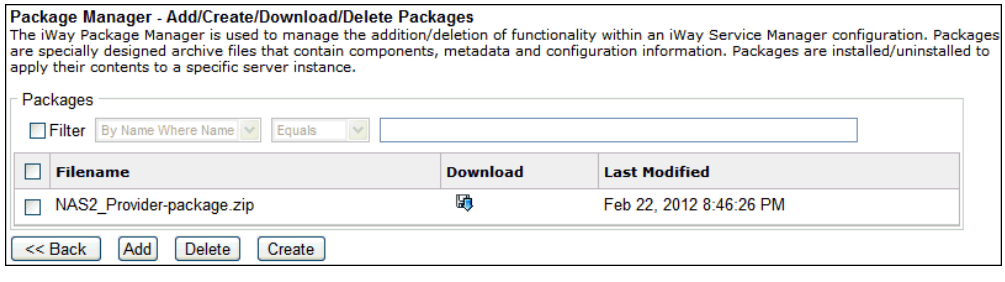


A message is displayed in the console indicating that the package was uploaded successfully, as shown in the following image.

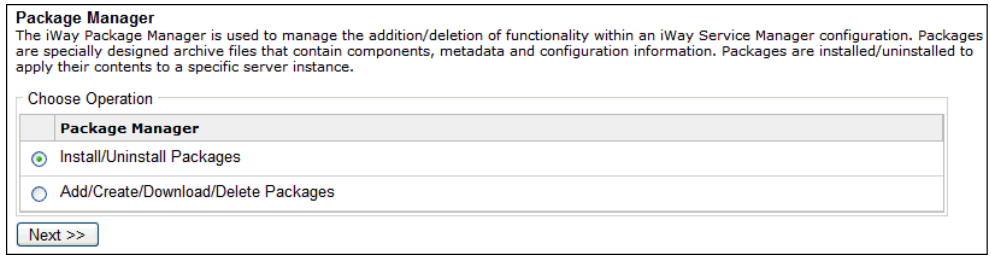


5. Click *Finish*.

You are returned to the Add/Create/Download/Delete Packages pane where the NAS2_Provider-package.zip file is listed.

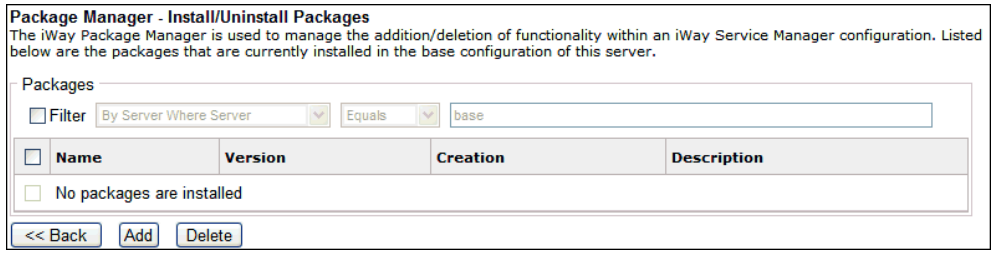


6. Once the package is added, click *Back* to navigate to the main Package Manager pane.



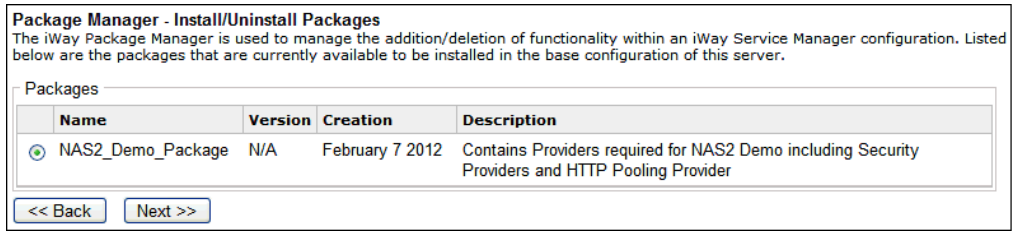
7. Select the first option, Install/Uninstall Packages, and then click *Next*.

   The Package Manager - Install/Uninstall Packages pane opens.



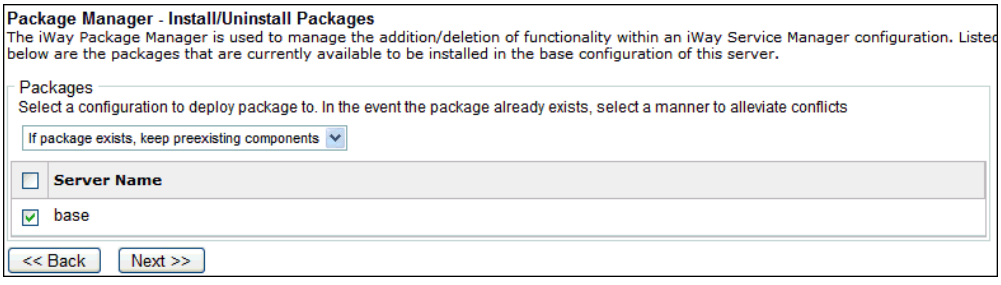
8. Click *Add*.

   The list of added packages is displayed, as shown in the following image.



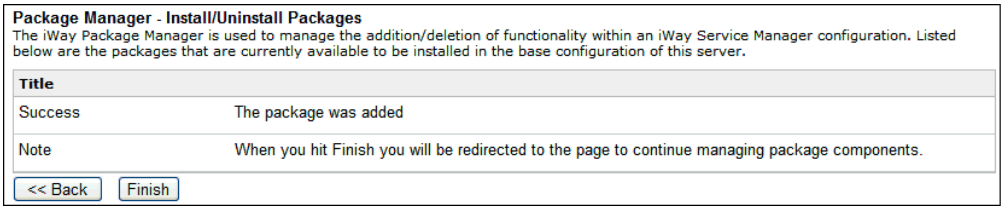
9. Select *NAS2_Demo_Package* and click *Next*.

The Server Name pane opens, as shown in the following image.



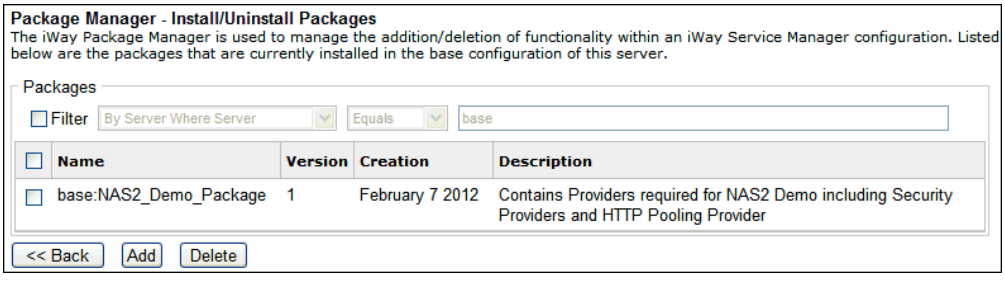
10. Select the *base* configuration and click *Next*.

    A message is displayed in the console indicating that the package was installed successfully, as shown in the following image.



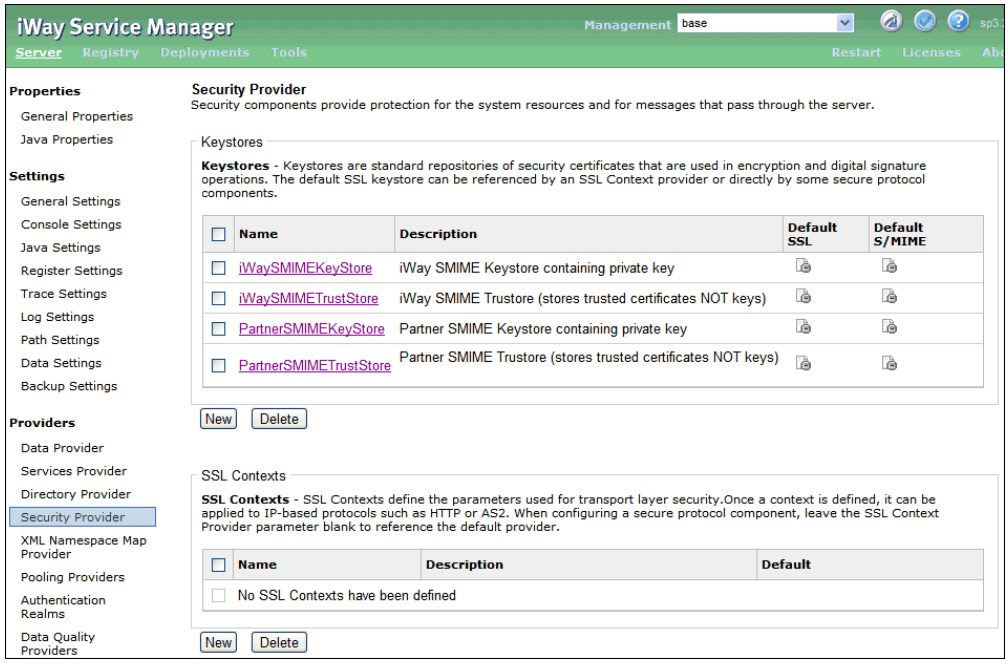
11. Click *Finish*.

    You are returned to the main Package Manager - Install/Uninstall Packages pane, which now lists the installed package.



12. Restart iWay Service Manager.

13. Select *Security Provider* in the left pane of the Server screen under the Providers section, as shown in the following image.



Notice that the four security providers (keystores) included in the NAS2_Provider package.zip file are now available in the console:

- iWaySMIMEKeyStore
- iWaySMIMETrustStore
- PartnerSMIMEKeyStore
- PartnerSMIMETrustStore

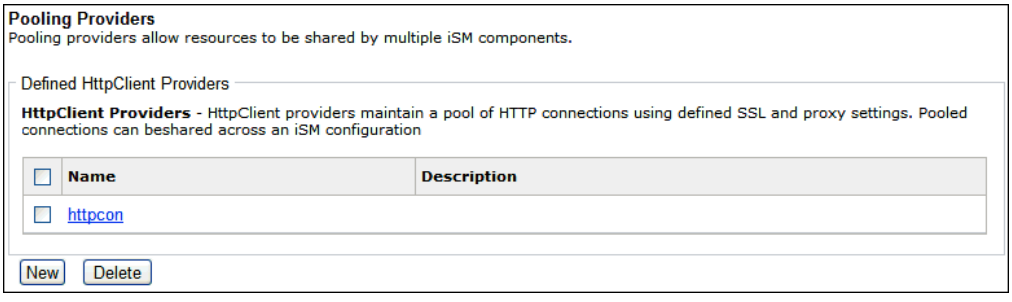
Note: These security providers include pre-configured references to keystore files. For example, the iWaySMIMEKeyStore security provider points to the following keystore file:

```
C:/keystore/iWaySMIMEKeyStore.p12
```

For your convenience and for reference purposes, a copy of the NAS2_Keystore_Files.zip file is attached to this document. The NAS2_Keystore_Files.zip file contains the keystore files that are associated by the deployed security providers.

For PDF-compatibility purposes, the file extension of the NAS2_Keystore_Files.zip file is temporarily renamed to **.zap**. After saving this file to your file system, you must rename this extension back to **.zip** before the keystore files can be extracted from this archive. After the files are extracted, create a keystore folder on your C drive and copy the keystore files into this folder.

14. Select *Pooling Providers* in the left pane of the Server screen under the Providers section, as shown in the following image.



Notice that the HTTP connection pool provider (httpcon) included in the NAS2_Provider package.zip file is now available in the console.

Chapter 3

Configuring NAS2 Channels

This section describes how to import archives, work with channels, and test their results.

In this chapter:

Importing Archives

This section describes how to import archives using the iWay Service Manager Administration Console.

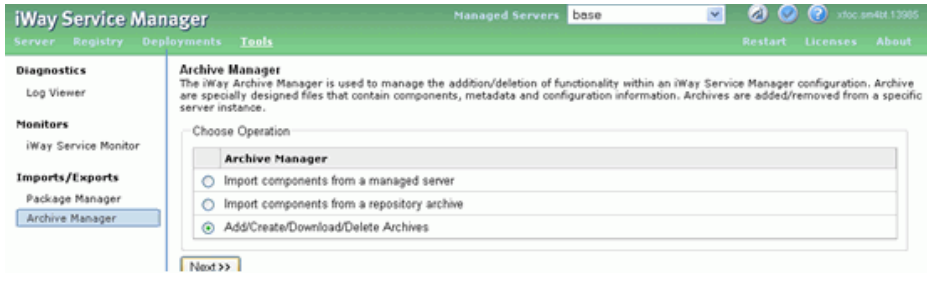
Procedure: How to Import Archives

To import archives:

1. Navigate to *Tools, Archive Manager* in the iSM console.

   The Archive Manager screen is displayed.

2. Select the third option, Add/Create/Download/Delete Archives, and then click *Next*, as shown in the following image.



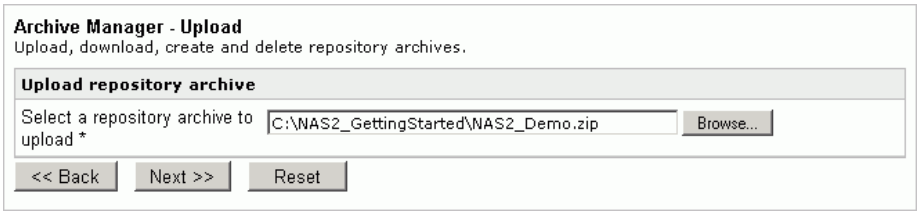
   The Add/Create/Download/Delete Archives pane is displayed.

3. Click *Add* and browse to the NAS2_Demo.zip archive on your file system.

Note: For your convenience and for reference purposes, a copy of the NAS2_Demo.zip archive is attached to this document. The NAS2_Demo.zip archive contains all the preconfigured iWay components (for example, listeners, process flows, channels, etc.) that are used and referenced in this NAS2 configuration.

For PDF-compatibility purposes, the file extension of the NAS2_Demo.zip archive is temporarily renamed to **.zap**. After saving this file to your file system, you must rename this extension back to **.zip** before it can be imported to iWay Service Manager.

4. Click *Next*, as shown in the following image.

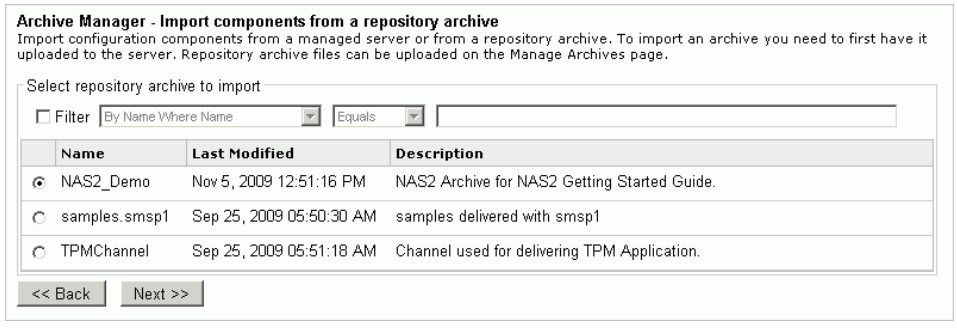


5. Once the archive is added, select *Back* to navigate back to the Add/Create/Download/Delete Archives pane.

6. Once back at the Add/Create/Download/Delete Archives pane, select the second option, Import components from a repository archive, and click *Next*, as shown in the following image.

The Import components from a repository archive pane is displayed.

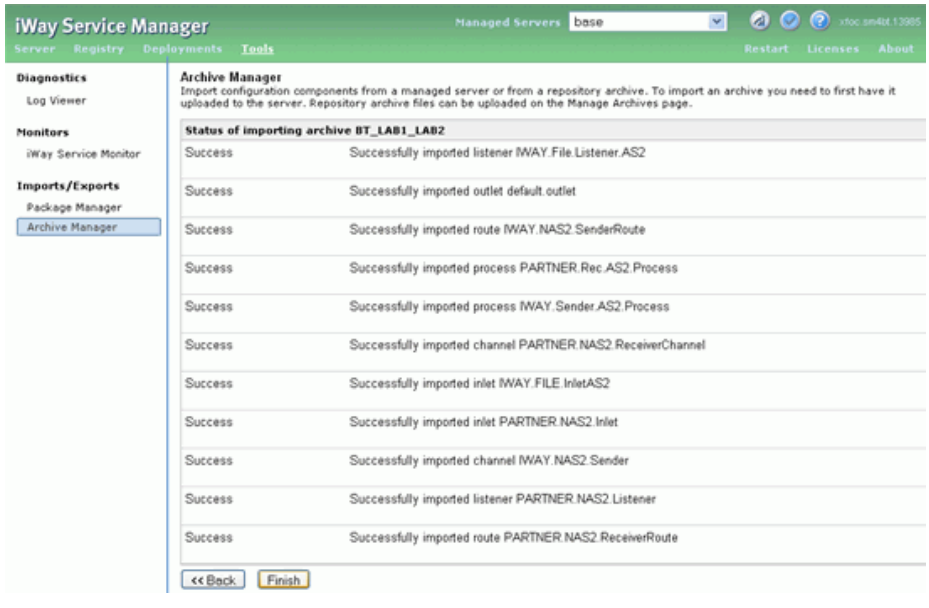
7. Select NAS2_Demo and click *Next* to add the components, as shown in the following image.



8. Check all items that are displayed and click *Next* to import them.

   A pane displaying the outlets that were imported is displayed.

9. After all the outlets have been imported successfully, click *Finish*, as shown in the following image.

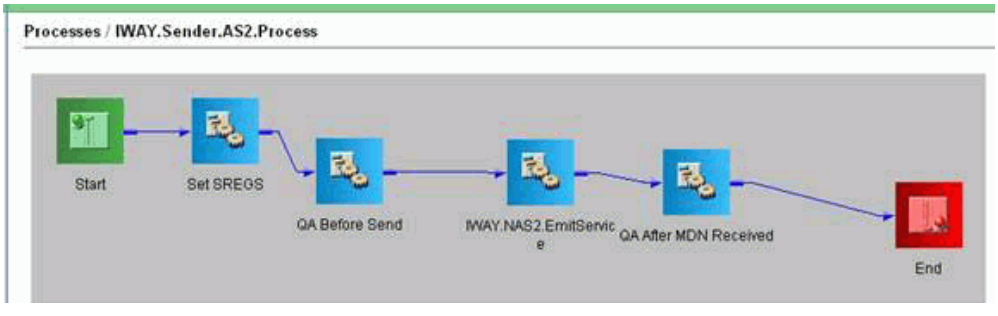


Sender (iWay) Channel Overview

This section describes in detail all the components for the iWay Sender channel. The channel uses a file listener to pick up files from the input directory and then sends them over AS2 protocol to the partner. The partner then responds with the MDN. For details on the configuration parameters, you can import the process flow into iWay Integration Tools (iIT) Designer and view the configuration as it is extensive. This section only provides a short overview of the process.

IWAY.NAS2.Sender Channel

- **IWAY.FILE.InletAS2:** Picks up files to be send to partner
  - Input Path = C:\NAS2_Demo\Input
  - Destination = C:\NAS2_Demo\Output
  - Suffix In = as2
- **IWAY.Sender.AS2.Process:** Uses NAS2 Emit Service to send file to partner



  - Start
  - Set SREGS: Sets HDR special registers within the namespaces to be used by the NAS2 Emit Service.
  - QA Before Send: Outputs a file to C:\NAS2_Demo\Output\IWAY \QA_before_NAS2_emit_*.txt to view the set Special Registers before the actual Emit via AS2.
  - IWAY.NAS2.EmitService: NAS2 Emit service is configured to send a signed document to http://localhost:5555. It uses the registers within the namespace created by the Set SREGS node.

- QA After MDN Received: Outputs a file to C:\NAS2_Demo\Output\IWAY \QA_after_MDN_recieved_*.txt showing the Special Registers after the NAS2 Emit has been completed. This will show the new mapping for the namespaces and will show that the original special registers are preserved.
- End

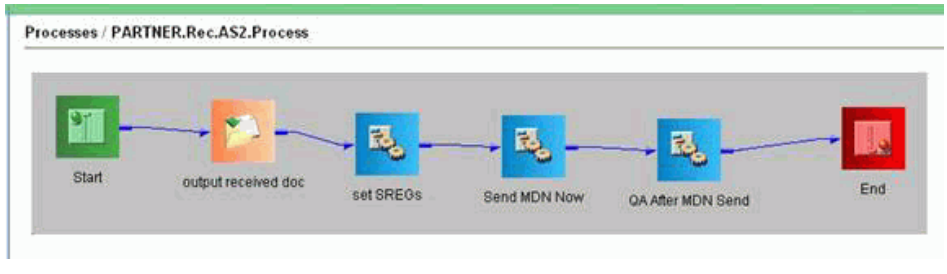
- default.outlet

Receiver (Partner) Channel Overview

Partner will receive the incoming messages on the NAS2 listener and will verify the signature. Once the signature is verified, the message will be processed by the Process Flow. The process flow will call the Send MDN Now service (com.ibi.agents.XDMDNSendNowAgent) to send an MDN after it is done processing (Delayed MDN Feature). It will also demonstrate the use of namespaces to propagate information and modify MDN before emit. To view the detailed process flow configuration, you can import the process flow into iWay Integration Tools (iIT) Designer. This section only provides a brief overview of the process.

IWAY.NAS2.Sender Channel

- PARTNER.NAS2.Listener: Receives documents on port 5555
- PARTNER.Rec.AS2.Process: Processes the document and sends back a delayed MDN



- Start
- Output Received doc: Writes the payload document to the directory C:\NAS2_Demo \Output\PARTNER\received_as2_doc_*.xml
- Set SREGS: Sets HDR special registers within the namespaces to be used by the Send MDN Now Node
- Send MDN Now: This service has no parameters as it uses the configured namespace registers from the listener configuration and from the Set SREGs node.

- QA After MDN Send: Outputs a file to C:\NAS2_Demo\Output\PARTNER \QA_After_MDN_isSend_*.txt showing the Special Registers after the MDN has been send. This will show that the received registers were mapped to the corresponding namespaces configured on the NAS2 Listener as well as the set registers for the set SREGs node.
- End

- default.outlet

Testing and Viewing Results

This section describes how to test and view the results.

Procedure: How to Test the AS2 Communication Process

To test the AS2 Communication Process:

1. Navigate to *Registry, Channels* in the iSM console.

   The Channels pane is displayed.

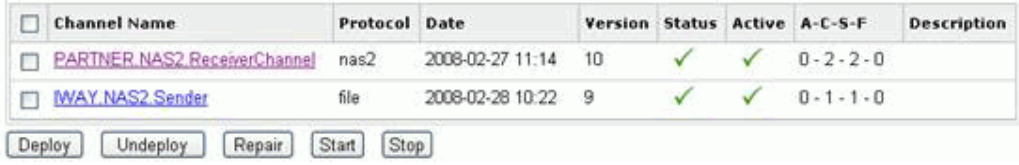
2. Select both IWAY.NAS2.Sender and PARTNER.NAS2.Receiver, then click *Build*, as shown in the following image.

3. Once the build completes, navigate to *Deployments, Channels*.

   The Channel Deployment pane is displayed.

4. Click *Deploy*.
5. Select *IWAY.NAS2.Sender* and then click *Deploy*.
6. Once IWAY.NAS2.Sender is deployed, repeat the same process for *PARTNER.NAS2.Receiver*.

7. After deploying the channels, select them both and click *Start*, as shown in the following image.



8. Drop any XML document to the C:\NAS2_Demo\Input\ location.

   The input_doc.xml document is provided as sample input in the same folder.

9. Copy the document and give it a .as2 extension.

   Since the File Listener on the Sender is listening for the .as2 extension, if it is given that extension, it will be picked up and processed.

   The following files will be displayed once everything is completed:

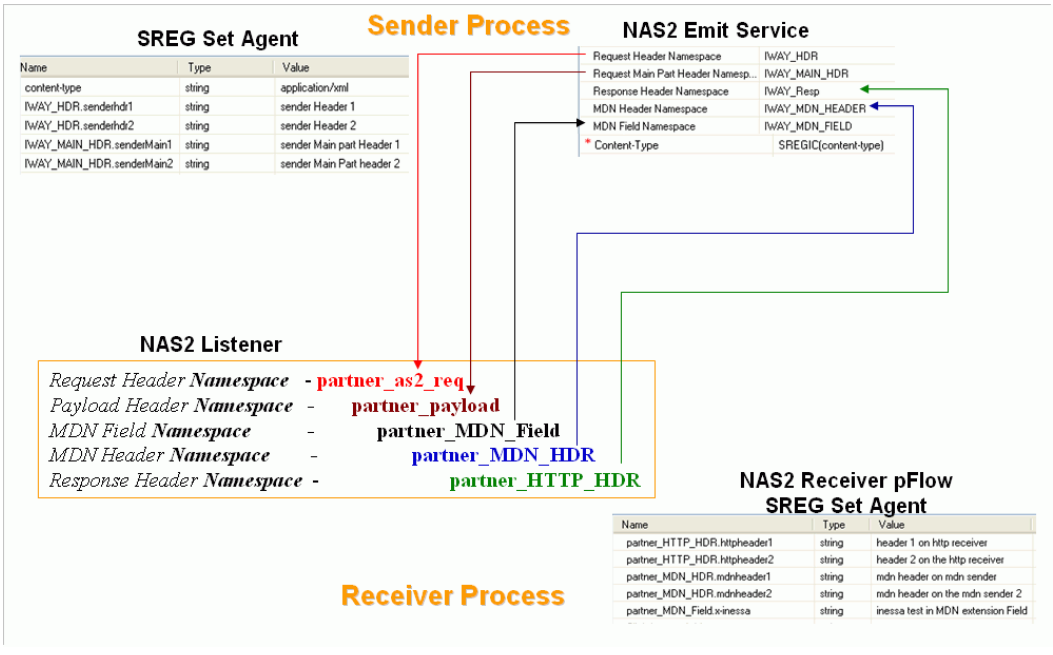
   - C:\NAS2_Demo\Output\IWAY\MDN_***.txt
   - C:\NAS2_Demo\Output\IWAY\QA_after_MDN_recieved_***.txt
   - C:\NAS2_Demo\Output\IWAY\QA_before_NAS2_emit_***.txt
   - C:\NAS2_Demo\Output\PARTNER\QA_After_MDN_isSend_***.txt
   - C:\NAS2_Demo\Output\PARTNER\received_as2_doc_***.xml

Reviewing Registered Namespaces Within the Output Files

This section provides a namespace mapping overview as well as a description of the multiple files that are included in the NAS2_Demo.zip archive.

Namespace Mapping Overview

The following diagram illustrates how the special register(SREG) namespaces are mapped from the Sender to the Receiver channel.



The SREG services on each side (sender/receiver) show how you would add/define additional registers, which you want to send using the request/response namespaces. In the Sender Process, the IWAY_HDR namespace is defined with two registers for the header. These two registers will be sent as part of the Request Header Namespace and will be received by the listener as part of Request Header Namespace (partner_as2_req). Two registers are also defined for the IWAY_MAIN_HDR namespace. These two registers will be sent as part of the Main Part Header Namespace and will be received by the Payload Header Namespace on the listener side. The listener in turn has its own registers configured for an MDN message which is sent back. These registers will be mapped, as shown by the arrows, and will be processed by the sender who is receiving the MDN. The details of the mappings can be seen in the Output Files as discussed in the following section.

Sender - Headers Within Received MDN (MDN***.txt)

Look for the section below in the MDN file. You can see “x-inessa” as one of the extension fields set through an SREG when generating a delayed MDN.

```
Reporting-UA: AS2 Server
Original-Recipient: rfc822, IWAY
Final-Recipient: rfc822, IWAY
Original-Message-ID:
<16055108.191204212785512.JavaMail.ig10588@DELL-IG>
Received-content-MIC: 6Qq5z1aghnSJ1WhbMTHN2g2Aulc=, SHA1
Disposition: automatic-action/MDN-sent-automatically; processed
x-inessa: inessa test in MDN extension Field
```

Sender - Headers Before Sending Message (QA_before_NAS2_emit***.txt)

Sender sets up the headers IWAY_HDR and IWAY_MAIN_HDR to propagate to the Receiver. However, it also wants to preserve them when the MDN is received. Without the use of namespaces, the header SREGs would be overwritten.

```
IWAY_HDR.senderhdr1          =    sender Header 1
IWAY_HDR.senderhdr2          =    sender Header 2
IWAY_MAIN_HDR.senderMain1    =    sender Main part Header 1
IWAY_MAIN_HDR.senderMain2    =    sender Main Part header 2
```

Receiver - Headers After MDN Send (QA_After_MDN_isSend***.txt)

You can see that the headers from the Sender have been mapped to the Receiver Header corresponding namespaces. As such, making them available for the Receiver but not changing them in the global view and preserving the original values to be reused by the sender.

```
partner_as2_req.senderhdr1          =    sender Header 1
partner_as2_req.senderhdr2          =    sender Header 2
partner_payload.senderMain1         =    sender Main part Header 1
partner_payload.senderMain2         =    sender Main Part header 2
partner_MDN_Field.x-inessa          =    inessa test in MDN extension Field
partner_MDN_HDR.mdnheader1          =    mdn header on mdn sender
partner_MDN_HDR.mdnheader2          =    mdn header on the mdn sender 2
partner_HTTP_HDR.httpheader1        =    header 1 on http receiver
partner_HTTP_HDR.httpheader2        =    header 2 on the http receiver
```

Receiver - Headers After MDN Received (QA_after_MDN_received***.txt)

You can see that the Sender has preserved the original Headers that it had (red & blue). However, it has also received a new set of headers from the Receiver of the Message and those headers were mapped to the corresponding configured namespaces.

```
IWAY_MDN_FIELD.x-inessa                =    inessa test in MDN extension Field
IWAY_MDN_HEADER.mdnheader1             =    mdn header on mdn sender
IWAY_MDN_HEADER.mdnheader2             =    mdn header on the mdn sender 2
IWAY_Resp.httpheader1                  =    header 1 on http receiver
IWAY_Resp.httpheader2                  =    header 2 on the http receiver
IWAY_HDR.senderhdr1                    =    sender Header 1
IWAY_HDR.senderhdr2                    =    sender Header 2
IWAY_MAIN_HDR.senderMain1              =    sender Main part Header 1
IWAY_MAIN_HDR.senderMain2              =    sender Main Part header 2
```

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, the TIBCO logo, the TIBCO O logo, FOCUS, iWay, Omni-Gen, Omni-HealthData, and WebFOCUS are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.