



FREE eBook

LEARNING wso2esb

Free unaffiliated eBook created from
Stack Overflow contributors.

#wso2esb

Table of Contents

About	1
Chapter 1: Getting started with wso2esb	2
Remarks	2
Examples	2
Installation or Setup	2
Chapter 2: Logging in WSO2 ESB	3
Examples	3
Separate log files for each proxy service	3
Chapter 3: WSO2 ESB - Local entries	5
Remarks	5
Examples	7
Example to retrieve local entry from sequences	7
Credits	10

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [wso2esb](#)

It is an unofficial and free wso2esb ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official wso2esb.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with wso2esb

Remarks

The WSO2 Enterprise Service Bus (ESB) is a fast, lightweight, 100% open source, and user-friendly ESB distributed under the Apache Software License v2.0. WSO2 ESB allows system administrators and developers to conveniently configure message routing, mediation, transformation, logging, task scheduling, failover routing, load balancing, and more. It also supports transport switching, eventing, rule-based mediation, and priority-based mediation for advanced integration requirements. The ESB runtime is designed to be completely asynchronous, non-blocking, and streaming based on the Apache Synapse mediation engine.

WSO2 ESB is developed on top of the revolutionary WSO2 Carbon platform, an OSGi-based framework that provides seamless modularity to your SOA via componentization. This release also contains many new features and a range of optional components (add-ons) that can be installed to customize the behavior of the ESB. Further, any existing features of the ESB that are not required in your environment can be easily removed using the underlying provisioning framework of Carbon. In brief, WSO2 ESB can be fully customized and tailored to meet your exact SOA needs.

Examples

Installation or Setup

To setup WSO2 ESB do the following :

1. Download WSO2 ESB from <http://wso2.com/products/enterprise-service-bus/> and extract the archive
2. Set the JAVA_HOME and PATH environment variables as per your OS
3. Navigate to [ESB_HOME]/bin. If on Windows run wso2server.bat. If on Linux run wso2server.sh.

More details can be found at <https://docs.wso2.com/display/ESB490/Installing+the+Product>

Read Getting started with wso2esb online: <https://riptutorial.com/wso2esb/topic/4178/getting-started-with-wso2esb>

Chapter 2: Logging in WSO2 ESB

Examples

Separate log files for each proxy service

All proxy services events are logged in the wso2carbon log file (located in %CARBON_HOME%/repository/logs).

If you want, you can have separate log files for each proxy service. Keep in mind, though, that they will still be logged in wso2carbon log file as well.

To do so, you should change the log4j.properties file, located in the %CARBON_HOME%/repository/conf folder.

For logs rolling on **time basis**, add the following lines at the bottom of the log4j.properties file:

```
log4j.category.SERVICE_LOGGER.Name_Of_Your_Proxy=INFO, YOUR_PROXY_SERVICE_APPENDER
log4j.additivity.SERVICE_LOGGER.Name_Of_Your_Proxy=false
log4j.appender.YOUR_PROXY_SERVICE_APPENDER=org.apache.log4j.DailyRollingFileAppender
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.File=logs/Name_Of_Your_Proxy.log
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.datePattern='.'yyyy-MM-dd
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.layout=org.apache.log4j.PatternLayout
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.layout.ConversionPattern=%d{ISO8601} \[%X{ip}-
%X{host}\] \[%t\] %5p %c{180} %m%n
```

Note that the `datePattern` determines how often a new file will be created. `yyyy-MM-dd` means that a new log file will be created daily and the old one will be renamed with the respective date. If you want to roll the file hourly, the pattern should also include the hours: `yyyy-MM-dd-HH`. Note that the colon character `:` should not be used anywhere in the `datePattern`.

For logs rolling on **size basis**, use the following:

```
log4j.category.SERVICE_LOGGER.Name_Of_Your_Proxy=INFO, YOUR_PROXY_SERVICE_APPENDER
log4j.additivity.SERVICE_LOGGER.Name_Of_Your_Proxy=false
log4j.appender.YOUR_PROXY_SERVICE_APPENDER=org.apache.log4j.RollingFileAppender
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.File=logs/Name_Of_Your_Proxy.log
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.MaxFileSize=10MB
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.MaxBackupIndex=100
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.layout=org.apache.log4j.PatternLayout
log4j.appender.YOUR_PROXY_SERVICE_APPENDER.layout.ConversionPattern=%d{ISO8601} \[%X{ip}-
%X{host}\] \[%t\] %5p %c{180} %m%n
```

You can set the max file size and the back up index to the desired values.

Note that `YOUR_PROXY_SERVICE_APPENDER` is an alias and should be different for each proxy service.

The location of the log files: `logs/Name_Of_Your_Proxy.log` can be changed.

`logs/Name_Of_Your_Proxy.log` leads to `%CARBON_HOME%/logs/Name_Of_Your_Proxy.log`.

Read Logging in WSO2 ESB online: <https://riptutorial.com/wso2esb/topic/4937/logging-in-wso2-esb>

Chapter 3: WSO2 ESB - Local entries

Remarks

In WSO2 registry is a content store and a metadata repository.

In WSO2 products and particularly WSO2 ESB uses registry to store metadata, artifacts (WSDL, XSD etc.,) and other service configurations like endpoints, sequences etc.,

Registry in WSO2 ESB has three flavours.

- Local registry
- Configuration registry
- Governance registry

Local Registry: It can be used to store configurations and metadata that is specific to the ESB node. This cannot be shared across all ESB nodes in the cluster or peer ESB nodes.

```
The mount path of local registry is  /_system/local
```

The local registry entries or local entries are very useful to store Strings, URLs, XSD / XSLT which can be easily retrieved from a mediator. These local entries can be accessed from the mediators / sequences by the following entry.

```
<localEntry key="validate_schema">
```

The local entries can be accessed from the ESB management console by the following way and artifacts can be added easily.

WSO2

Enterprise Service Bus

Home

Manage

Main

Monitor

Configure

Tools

Services

List

Add

Proxy Service

Service Bus

Sequences

Inbound Endpoints

Scheduled Tasks

Templates

Endpoints

Message Processors

Local Entries

Message Stores

Priority Executors

APIs

Source View

Connectors

List

Add

Secure Vault Tool

Manage Passwords

Carbon Applications

List

Add

Modules

List

Add

Topics

Browse

Add

ESB Artifacts

List

Add

Shutdown/Restart

Registry

Home > Manage > Service Bus > Local Entries

Manage Local Registry Entries

Available Local EntriesAdd Local Entries

Select Local Entry type to be added

+ Add In-lined Text Entry	Text content that is l
+ Add In-lined XML Entry	XML content (XSLT, P
+ Add Source URL Entry	Hosted resource acco

Examples

Example to retrieve local entry from sequences

In WSO2 registry is a content store and a metadata repository.

In WSO2 products and particularly WSO2 ESB uses registry to store metadata, artifacts (WSDL, XSD etc.,) and other service configurations like endpoints, sequences etc.,

Registry in WSO2 ESB has three flavours.

- Local registry
- Configuration registry
- Governance registry

Local Registry: It can be used to store configurations and metadata that is specific to the ESB node. This cannot be shared across all ESB nodes in the cluster or peer ESB nodes.

The mount path of local registry is /_system/local

The local registry entries or local entries are very useful to store Strings, URLs, XSD / XSLT which can be easily retrieved from a mediator. These local entries can be accessed from the mediators / sequences by the following entry.

```
<localEntry key="validate_schema_XSLT">
```

The local entries can be accessed from the ESB management console by the following way and artifacts can be added easily.

WSO2 Enterprise Service Bus

Home

Manage

Main

- Services
 - List
 - Add
 - Proxy Service
- Service Bus
 - Sequences
 - Inbound Endpoints
 - Scheduled Tasks
 - Templates
 - Endpoints
 - Message Processors
 - Local Entries
 - Message Stores
 - Priority Executors
 - APIs
 - Source View
- Connectors
 - List
 - Add
- Secure Vault Tool
 - Manage Passwords
- Carbon Applications
 - List
 - Add
- Modules
 - List
 - Add
- Topics
 - Browse
 - Add
- ESB Artifacts
 - List
 - Add
- Shutdown/Restart

Registry

Home > Manage > Service Bus > Local Entries

Manage Local Registry Entries

Available Local EntriesAdd Local Entries

Select Local Entry type to be added

+ Add In-lined Text Entry	Text content that is l
+ Add In-lined XML Entry	XML content (XSLT, P
+ Add Source URL Entry	Hosted resource acco

<https://riptutorial.com/>

8

Read WSO2 ESB - Local entries online: <https://riptutorial.com/wso2esb/topic/7453/wso2-esb---local-entries>

Credits

S. No	Chapters	Contributors
1	Getting started with wso2esb	Community , maheeka
2	Logging in WSO2 ESB	Maria Deleva
3	WSO2 ESB - Local entries	Muralidharan.rade