

RTI Connex Integrator

Getting Started Guide

Version 4.5



Your systems. Working as one.



© 2012 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
March 2012.

Trademarks

Real-Time Innovations, RTI, DataBus, and Connex are trademarks or registered trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of their respective owners.

Copy and Use Restrictions

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form (including electronic, mechanical, photocopy, and facsimile) without the prior written permission of Real-Time Innovations, Inc. The software described in this document is furnished under and subject to the RTI software license agreement. The software may be used or copied only under the terms of the license agreement.

Technical Support

Real-Time Innovations, Inc.
232 E. Java Drive
Sunnyvale, CA 94089
Phone: (408) 990-7444
Email: support@rti.com
Website: <https://support.rti.com/>

Contents

1	Available Documentation	2
1.1	RTI Routing Service	2
1.2	RTI Routing Service Adapter SDK	2
1.3	RTI Real-Time Connect	3
2	Installing Connex Integrator.....	3
3	Understanding the Connex Architecture.....	4
4	Using Connex Integrator Effectively	6
4.1	Configuring RTI Routing Service	7
4.2	Configuring RTI Routing Service Adapter SDK	9
4.3	Configuring RTI Real-Time Connect.....	10

Welcome to Connex Integrator

Welcome to RTI® Connex™ Integrator! *Integrator* is a flexible integration infrastructure for building operational systems and bridging to IT applications, while requiring little or no modification to existing deployed applications. It is based on a proven foundation of technology and tools used in highly diverse systems, often with extreme real-time performance requirements.

Integrator extends the benefits of Service-Oriented Architecture (SOA) out to the operational edge of the network. The following are some of these key benefits.

- ❑ **Combine applications into systems.** Integrate existing deployed real-time and non-real-time applications with significantly reduced time, cost, and effort. Add or remove applications from an existing system, transparently. And carefully control which data is shared among applications and which remains securely within the boundaries of a single application.
- ❑ **Preserve performance and determinism.** Connect applications with confidence, knowing that the data routed between them will not impact the behavior of time- and resource-sensitive logic. Isolate private data and metadata, and control the timing of communication to avoid overwhelming slow consumers.
- ❑ **Govern interactions among applications.** Enforce strict contracts on which data can be produced and/or consumed, by whom, with what quality of service.
- ❑ **Upgrade incrementally.** Versioning is a consistent pain in distributed systems: you need to upgrade to a new version of one component or application, but it's not possible to upgrade an entire system of systems all at once. *Integrator* isolates version changes to the affected components and gives you control over how those changes get mediated to the rest of the system. And with its standards-based protocol and industry-leading track record of interoperability, you can even *upgrade Integrator itself* incrementally, without breaking your system.

- ❑ **Modularize your design over time.** It's not practical to rewrite your applications from scratch, but you may find that their existing implementation makes it hard to evolve them. *Integrator* can help you define subsystem boundaries that allow you to cleanly improve and extend existing applications without impacting existing functionality.
-

1 Available Documentation

Connex *Integrator* includes the following components:

- ❑ *RTI Routing Service*
- ❑ *RTI Routing Service Adapter SDK*
- ❑ *RTI Real-Time Connect*

This document describes how these components fit together. Detailed documentation is provided with each component.

1.1 RTI Routing Service

Routing Service is at the core of the *Integrator* product architecture. Its role is to connect the individual applications that comprise the system to the integration bus. The integration bus is in turn constructed by these *Routing Service* instances—although it can be administered centrally, the bus is physically distributed. For more information, see [Understanding the Connex Architecture \(Section 3\)](#).

1.2 RTI Routing Service Adapter SDK

Routing Service Adapter SDK provides a solution for interfacing with third-party protocols and messaging technology. It includes prebuilt adapters that can be used out of the box. It also provides service-provider interfaces (SPIs) to allow users to extend *Connex Integrator* themselves with additional adapters. Pre-built adapters include source code

so that they can be easily modified to meet system-specific requirements or serve as a template for quickly creating new adapters.

1.3 RTI Real-Time Connect

Real-Time Connect provides adaptation between the integration data bus and a relational database. It can be used standalone to provide a straightforward view of real-time data through SQL, or it can be used together with *Routing Service* to adapt to application-specific schemas.

2 Installing Connex Integrator

Connex Integrator is an extension to *Connex Messaging* or *Connex DDS*. It includes the following additional components:

- RTI Routing Service*
- RTI Routing Service Adapter SDK*
- RTI Real-Time Connect*

These are available for download from your RTI customer portal at <http://support.rti.com> and must be installed separately.

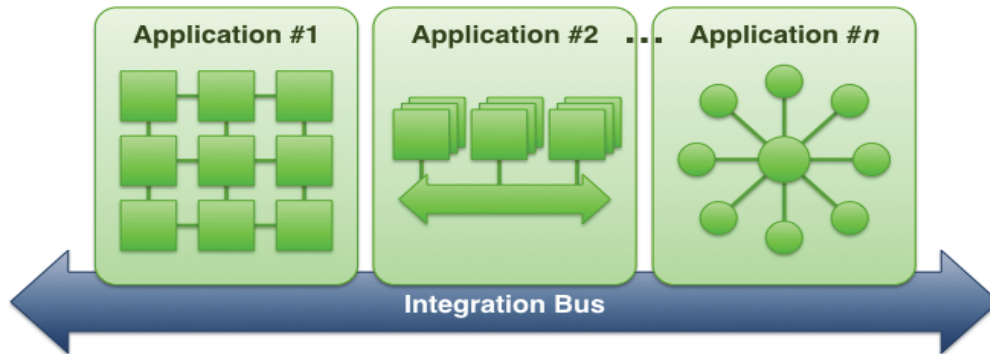
1. Install *Connex Messaging* (or *Connex DDS*) itself. Installation instructions are provided in the corresponding *Getting Started Guide: RTI_Connex_GettingStarted.pdf*.
2. Install *Routing Service*. Installation instructions are provided in the *Routing Service Getting Started Guide: RTI_Routing_Service_GettingStarted.pdf*.
3. Install *Routing Service Adapter SDK*. Installation instructions are provided in the *Routing Service Adapter SDK Installation Guide: RTI_Routing_Service_AdapterSDK_InstallationGuide.pdf*.
4. Install *Real-Time Connect*. Installation instructions are provided in the *Real-Time Connect Getting Started Guide: RTI_RTC_GettingStarted.pdf*. These instructions are customized to the particular database with which you intend to use *Real-Time Connect* (Oracle, MySQL, etc.).

Routing Service and *Real-Time Connect* can be installed in either order, but *Routing Service Adapter SDK* must be installed after *Routing Service* itself.

3 Understanding the Connext Architecture

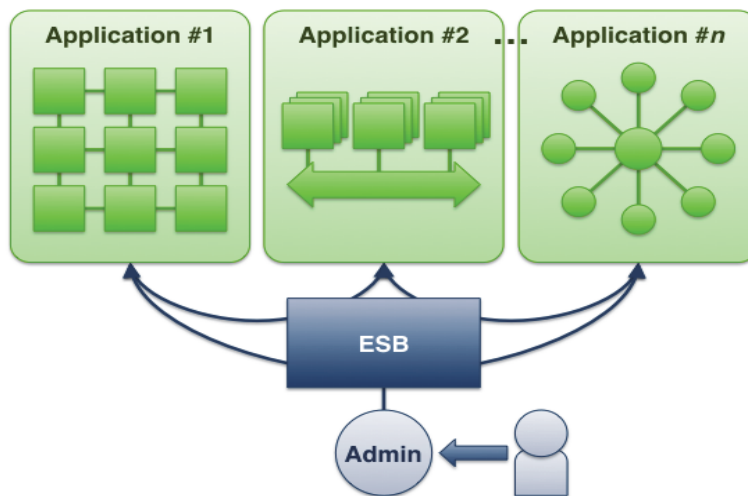
Consider some number of applications, which are developed, deployed, and maintained independently of one another—probably by independent groups with their own responsibilities, priorities, and timetables. We want to integrate them together into a coherent system.

Figure 3.1 Abstract integration Architecture



With a traditional ESB, the topology would look like this:

Figure 3.2 Integration using a Traditional ESB

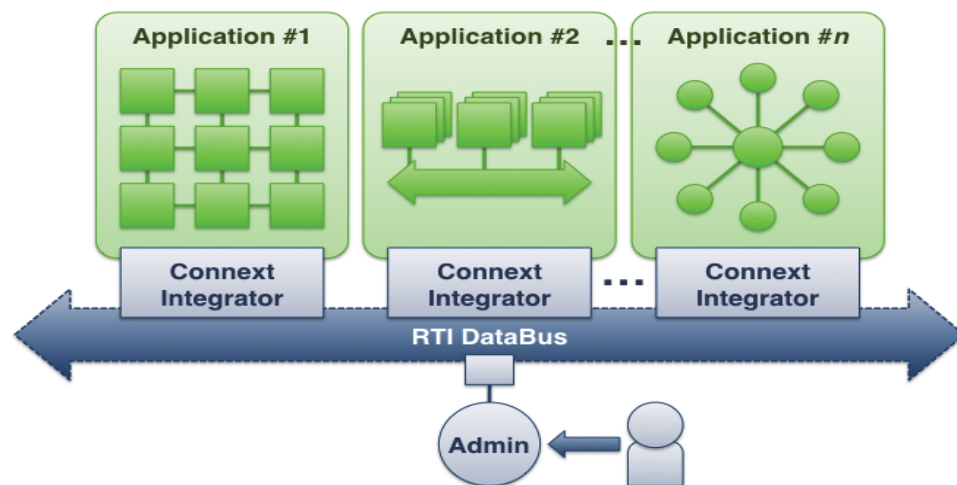


That is, a central service would be aware of every application, and it would be responsible for performing one-to-one adaptation between each pair of them. This approach works when the number of applications (in their various versions) is small, the amount of data that applications need to exchange is limited, and a centralized location exists at which to manage the integration infrastructure. In traditional IT applications, these conditions often *are* true. However, in systems that are physically distributed or must deal with physical processes or devices, they typically are not—and as soon as those assumptions break down, so does the integration.

- ❑ As the number of applications “ n ” increases, the amount of effort to integrate them pair-wise increases with the *square* of n .
- ❑ When applications share high-rate data, such as from a sensor, these data streams can overwhelm the broker, which becomes a bottleneck. And even if the broker can keep up, it can become a source of jitter for applications that require deterministic behavior.
- ❑ When applications are mobile—for example, when they may run in handheld or mounted platforms—there may be no practical place to run centralized infrastructure.

Connex Integrator is different. It decentralizes and optimizes the integration infrastructure, just like *Connex Messaging* decentralizes and optimizes the messaging infrastructure within an application. At the same time, it preserves an administrator’s ability to monitor and manage the infrastructure from a single location.

Figure 3.3 Integration using Connex Integrator

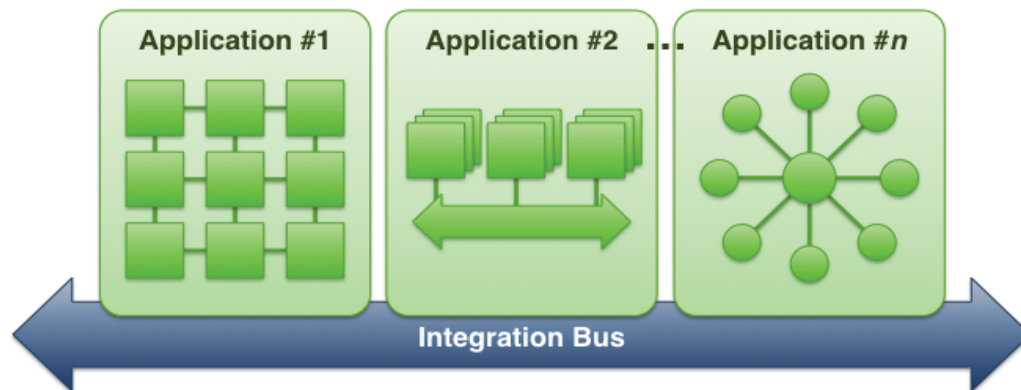


Instances of the *Connex Integrator* software infrastructure services construct the integration bus among themselves. A single instance (or a small group of redundant copies) is responsible for adapting those integration flows to the needs of a particular application. Other instances do not need to know anything about that application; they have their own responsibilities. Thus, the work of integrating, upgrading, or removing a given application remains constant. And because the work of integration is decentralized, there are no artificial performance bottlenecks or single points of failure.

4 Using Connex Integrator Effectively

The best practice for using *Connex Integrator* is to consider integration as a separate activity from application development—an activity with its own lifecycle: design, implementation, deployment, and evolution.

Figure 4.1 **Configurable Elements**



In the figure above, there are $(n + 1)$ elements: the n applications plus the integration infrastructure itself. Each has its own concerns, constraints, and responsibilities.

In *Connex Integrator*, the integration bus is an instantiation of the *RTI DataBus*[™], and the participants on that bus are instances of *RTI Routing Service*. As always, the *DataBus* is *data-centric*—it operates on distinct data elements and the schemas and quality-of-service (QoS) policies attached to them. Therefore, your first task when using *Integrator* is to decide what those data elements, schemas, and policies will be—we'll call the collection of those things the *integration data model*. You can find more information about this process in the *RTI Core Libraries and Utilities Getting Started Guide* and *User's Manual*.

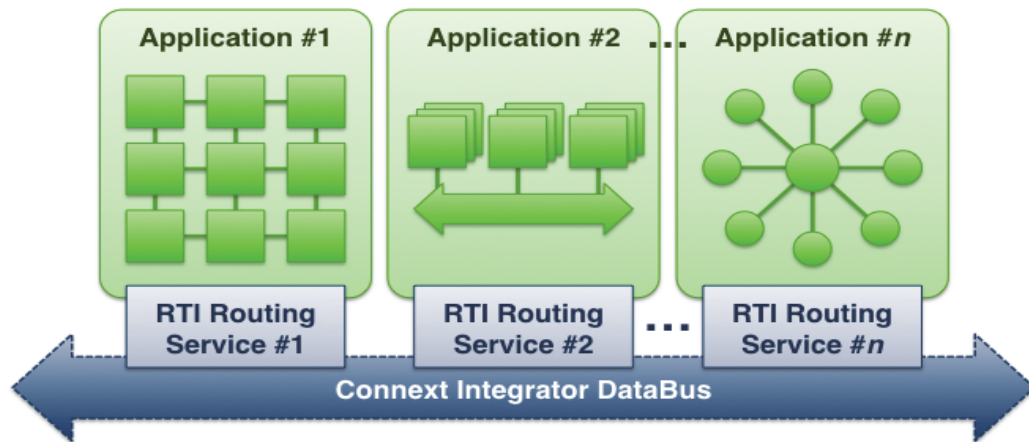
Each application, in turn, will have its own way of exchanging data. It may use its own middleware or protocol to communicate. It may have a unique data model or message set. For an application to connect to the integration bus, that application-specific communication must be translated into the data model of the integration bus. The following sections introduce the *Connex Integrator* components and how to use them to accomplish this task.

4.1 Configuring RTI Routing Service

RTI Routing Service is the backbone of *Connex Integrator*.

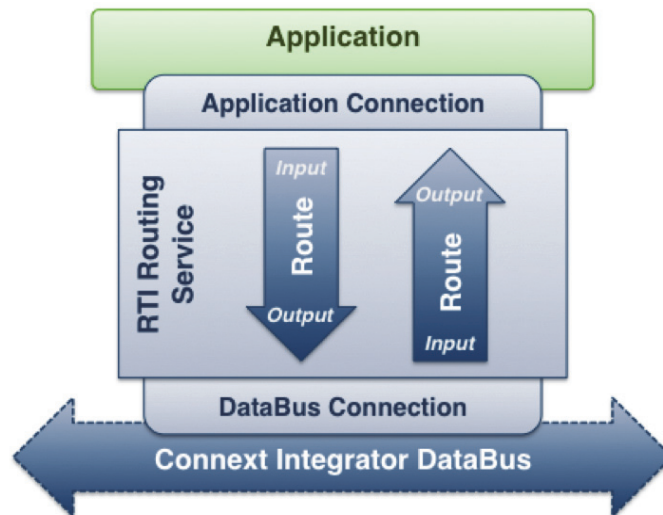
One instance of *Routing Service* (or a small number of redundant instances, for fault tolerance) is typically responsible for connecting a single application to the integration *DataBus*.

Figure 4.2 Using RTI Routing Service



The job of the Routing Service instance(s) assigned to a particular application is to trans-

Figure 4.3 **Configuring RTI Routing Service**



late between the data model or message set of that application and the integration data model. Through its configuration, you will instruct it how to do that. That configuration organizes data flows in terms of *routes* between *connections*.

When you configure an instance of Routing Service to connect a particular application to the integration *DataBus*, you will configure two categories of routes: those that flow from the application into the integration *DataBus* and those that flow in the opposite direction. As data flows through the route, there are two kinds of translation that take place:

- 1. Data content transformation.** Unless the applications you're integrating were designed in very close collaboration with one another, it is likely that the data and message formats used in a particular application will be different from those used in the others, and therefore different from the ones you chose for the integration *DataBus*. *Routing Service* provides a Transformation SDK to allow you to plug in whatever data content transformations you require.
- 2. QoS mediation.** The integration *DataBus* is QoS-aware: every piece of data it exchanges is governed by explicit policies that declare how that data moves through the system and how it changes over time. On the side of the route facing the integration *DataBus*, you will specify the policies you've chosen for the inte-

gration data model. If the application is also based on *Connex* technology, you will also have a set of similar policies configured there, although the specific policy values may well be different.

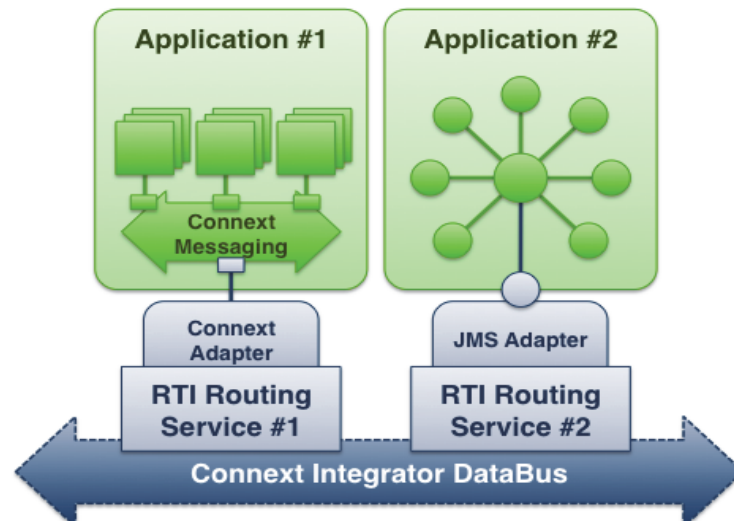
You can find more information about configuring *Routing Service* in the *RTI Routing Service Getting Started Guide* and *User's Manual*.

4.2 Configuring RTI Routing Service Adapter SDK

RTI Routing Service Adapter SDK provides adaptation to lightweight messaging-style technologies. It includes out-of-the-box adapters for JMS, bare sockets, and even flat files. It also includes SPIs that allow you to develop your own adapters. The prebuilt adapters include source code so that you can customize them for the particular needs of your applications or use them as templates for the adapters you develop yourself.

Within the *Routing Service* configuration, a connection is associated with the use of a plug-in communication *adapter*, configured as appropriate for a particular application. (The connection to the integration *DataBus* has an “adapter” as well, although this adapter is well known to *Routing Service* ahead of time; it is built in and designed for easy configuration.)

Figure 4.4 Using the RTI JMS Adapter



For example, suppose that one of your applications is using JBOSS Messaging for its internal communication. You will define a connection to that middleware using the RTI JMS Adapter included in the *RTI Routing Service Adapter SDK* bundle, and you will configure that connection with the appropriate properties for that application. Then you

will define routes between the JBOSS topics and queues used by the application and the topics you defined as part of the integration data model. At run time, *Routing Service* will appear to the JBOSS Messaging broker as simply one more client, which will produce and consume messages like any other.

The *Routing Service Getting Started Guide* contains a number of examples and exercises to help you become comfortable with these tasks. The *Routing Service User's Manual* contains detailed configuration instructions.

4.3 Configuring RTI Real-Time Connect

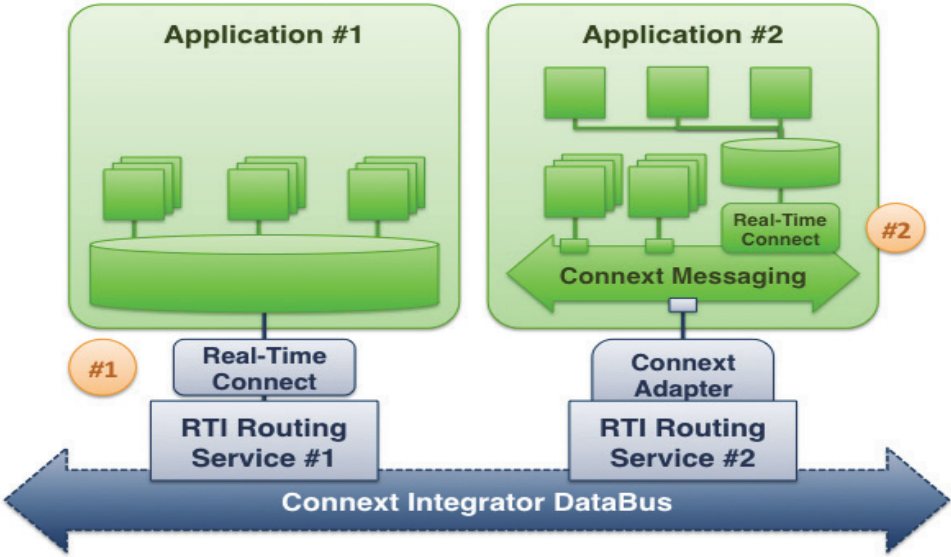
As the *Routing Service Adapter SDK* connects the data flowing along the integration *DataBus* with application messaging protocols, *Real-Time Connect* connects it with the data that is (relatively) at rest in a relational database. When new data values are published on the bus, it inserts those into the database. When the database changes, it publishes those new values unto the bus.

When integrating applications to create systems, use *Real-Time Connect* alongside *Routing Service* as an “adapter” between an application and the integration *DataBus*. As always, the *Routing Service* instance will be primarily responsible for translating between the integration data model and the data model used by the application. *Real-Time Connect* will take that data into and out of the database.

You can also use *Real-Time Connect* within an application, as a view onto the *DataBus*. If you are using *Connexx Messaging* or *Connexx DDS* for communication within an application, you can connect an instance of *Real-Time Connect* directly to the application's *DataBus*. This will allow you to write more time-critical portions of your application using the DDS or JMS APIs provided by *Connexx* while writing analytical components using

the SQL interface provided by the database. However you access the data, you will see the same data model reflected.

Figure 4.5 Configuration Scenarios for RTI Real-Time Connect



For more information about *Real-Time Connect*, see the *RTI Real-Time Connect Getting Started Guide* and *User's Manual*.

