

# About the Data Repository

This chapter discusses the Data Repository. In this section:

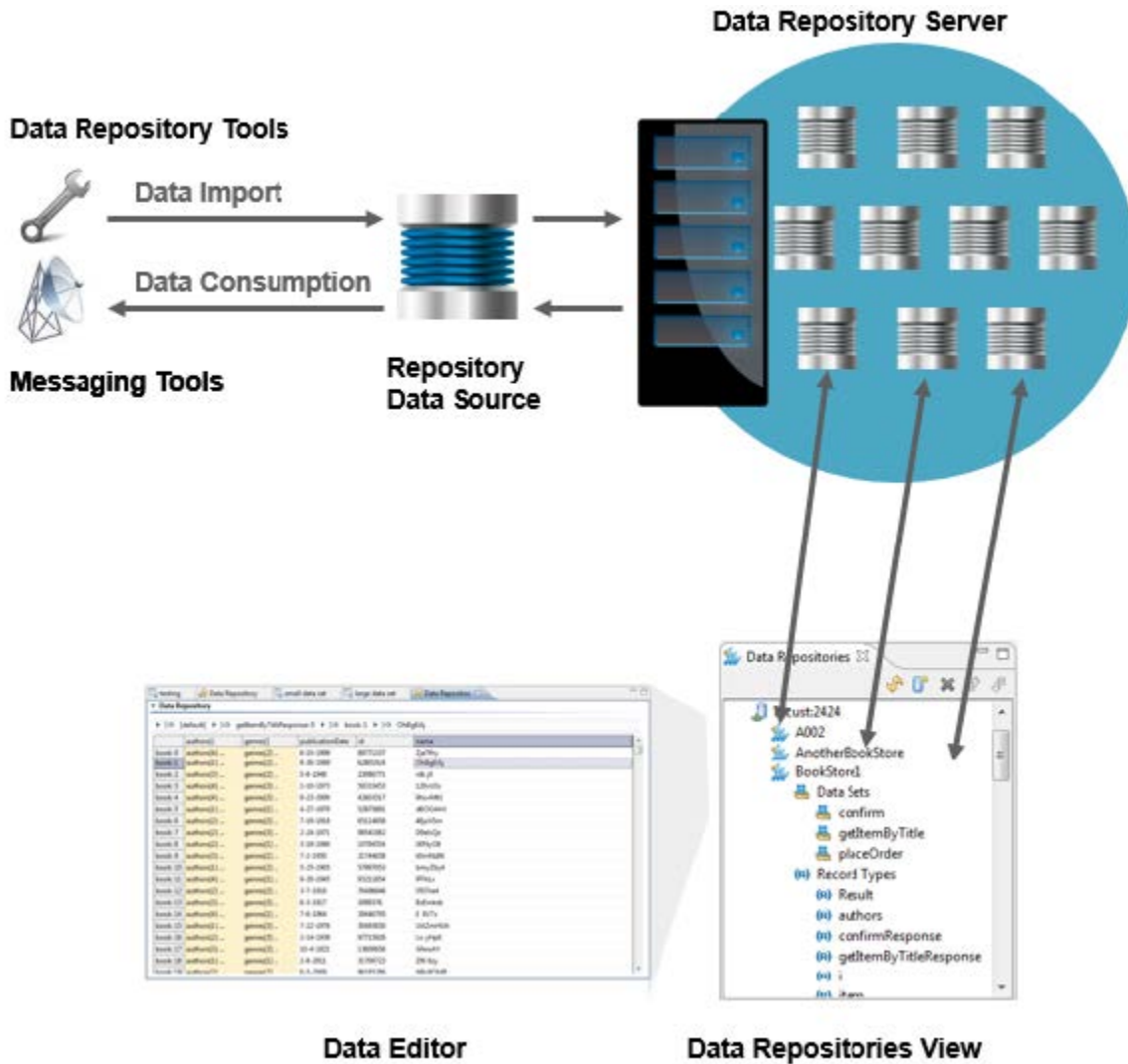
- [Overview](#)
- [Concepts and Terminology](#)
- [Structure Requirements](#)

## Overview

Parasoft's Data Repository is designed to help teams define, extend, and review large and/or hierarchical data sets for use in Parasoft messaging tools. The same data sets can be used for service virtualization in Parasoft Virtualize and for API testing in Parasoft SOAtest. They can also be used from Parasoft CTP.

Once a repository is established on a data repository server, it can be populated from existing data sources and/or updated manually. Through the graphical representation of hierarchical data, you can review and extend the repository structure and contents. Records from one data set can be referenced in other data sets to simplify editing and management of large data sets with a high level of data reuse.

The following is a very brief overview of the key components involved in working with Parasoft's Data Repository.



Your group can deploy any number of *Data Repository Servers*. Each Data Repository Server hosts any number of repositories. A repository can be populated by importing data from an existing data source; you can also manually define, extend, and edit it using the graphical *Data Repositories View* and *Data Editor*.

The *Data Repositories* view is your main control point for managing servers and adding, deleting, editing, and reviewing repositories on each of those servers. Each repository's hierarchy is represented graphically, and this graphical representation links to a *Data Editor* that simplifies extending and browsing large, complex data sets.

The data stored in repositories can be consumed by SOAtest and Virtualize messaging tools via connection to a *Repository Data Source*. This allows you to take advantage of Parasoft's form parameterization platform, as well as data group functionality.

## Concepts and Terminology

When working with the Data Repository, it's helpful to understand the following key concepts and terms:

- [Data Repository Server](#)
- [Data Repository](#)
- [Data Set](#)
- [Record Type](#)
- [Repository Connection](#)
- [Repository Data Source](#)
- [Data Repositories View](#)
- [Data Editor](#)

### Data Repository Server

The Data Repository Server is the server that stores the data. You can have one server shared by multiple teams, or you can deploy multiple servers.

A "sandbox" Data Repository is embedded within each Virtualize or SOAtest installation. This allows you to quickly create parameterized assets for use on your local system.

Additionally, remote Data Repository Server can be installed on any systems; a remote server does not need to be installed on a system where you have Parasoft Virtualize or Parasoft SOAtest installed. This server can be shared across teams or divisions.

Each Data Repository Server can host multiple Data Repositories.

### Data Repository

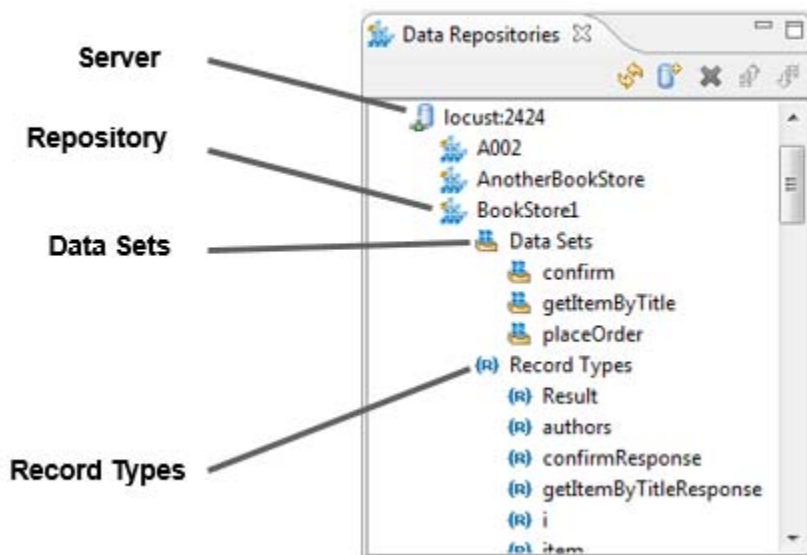
A Data Repository is a group of related data sets that are stored on a Data Repository Server. Each repository has one or more data sets which share a distinct library of record types. Import and export occurs at the data repository level.

The scope of a data repository is entirely up to you. It can be extremely broad, or very specific. As you structure repositories, consider the record types that will need to be shared as well as the export/import and backup/restore cycles. For instance, you might want to create one repository for services that share the same record types, and a different one for services that use a very different set of record types. For another example, assume you have two teams: one that will be frequently modifying data then restoring the original data, and another set of teams that needs continuous access to a stable version of that same data set. In that case, it would probably make sense to have 2 different data repositories.

Not all data repositories have to be represented in every team member's Data Repositories view. For example, assume your division is sharing a Data Repository Server that hosts 20 repositories. Members of Team A might only be working with 3 of those repositories, so they would need only those 3 repositories connected to their Data Repositories view. Another team might choose to connect to 1 of those repositories along with 5 additional repositories.

### Data Set

A data set is a subset of data records. The various data sets on a repository all tap into the same library of record types. For instance, if multiple data sets in a data repository use the record type "book," they will all access the same instance of the book record type. Each data set always has at least one record type.



## Record Type

A record type has a name, a certain number of columns (a.k.a. fields), and a hierarchy defined. The scope of record types is at the repository level (e.g., the same "book" record type definition applies across all data sets that use the "book" record type).

Data Repository						
item: 0 ▶ Great for Java beginners						
	id	authors:[R]	title	price	description	IS
item: 0	2	authors:[1]	Java How to ...	76.0	Great for Java ...	0
item: 1	1	authors:[1]	C++ How to ...	99.99	One of the be...	0
item: 2	4	authors:[1]	Linux Admini...	49.99	Provides tech...	0

## Repository Connection

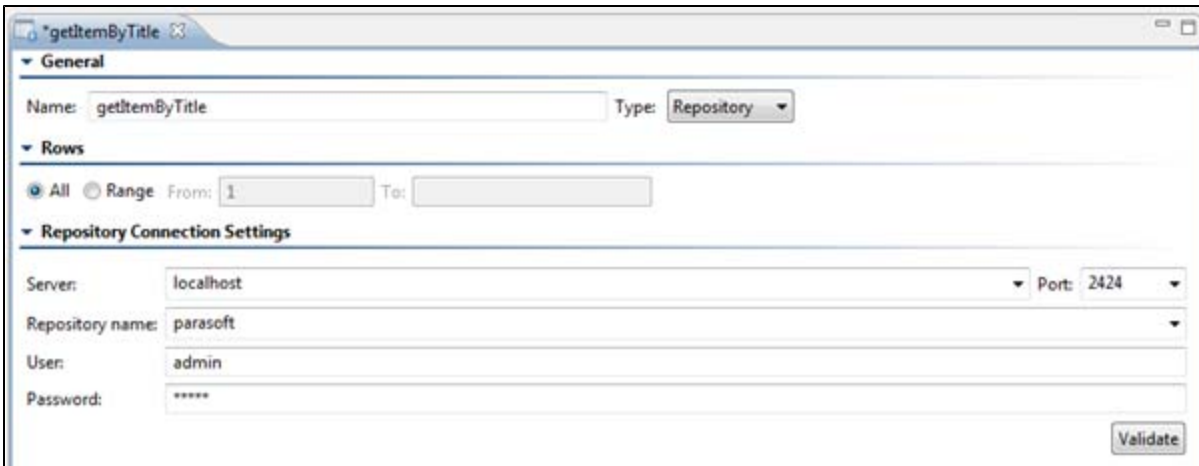
A Repository Connection adds a Data Repository to the Data Repositories view, which helps you review and edit the structure and contents.

The screenshot shows the 'Repository Connection Settings' dialog box. The title bar reads 'people on localhost'. The settings are as follows:

- Host: localhost
- Port: 2424
- Repository name: people
- User: admin
- Password: \*\*\*\*\*

## Repository Data Source

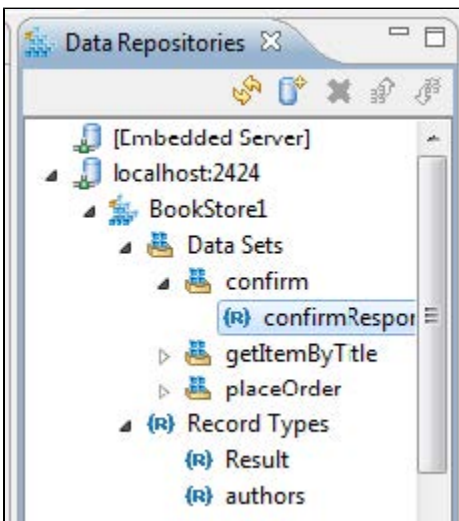
A repository data source is a "wrapper" for the repository data source that allows you to consume it in Parasoft messaging tools—in the same manner as you would consume data from an Excel data sheet, CSV file, or other supported data source. Through the data source wrapper, you can segment what repository data you want to use for a given suite (for instance, only certain rows). You can also create "data groups." With data groups, you can group similar sets of data (such as development environment test data and load/performance test data), then easily switch which data set is used at any given time— without having to edit the tool or data source configurations.



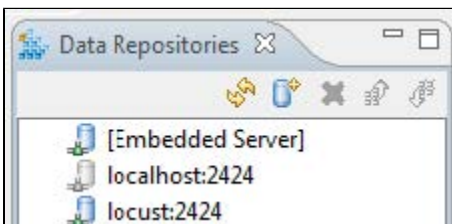
## Data Repositories View

The Data Repositories view provides a UI for reviewing and editing data repositories. It can be used to create new repositories as well as to view and navigate through existing ones.

Double-clicking a node in the Data Repositories view opens the related records in the Data Editor.



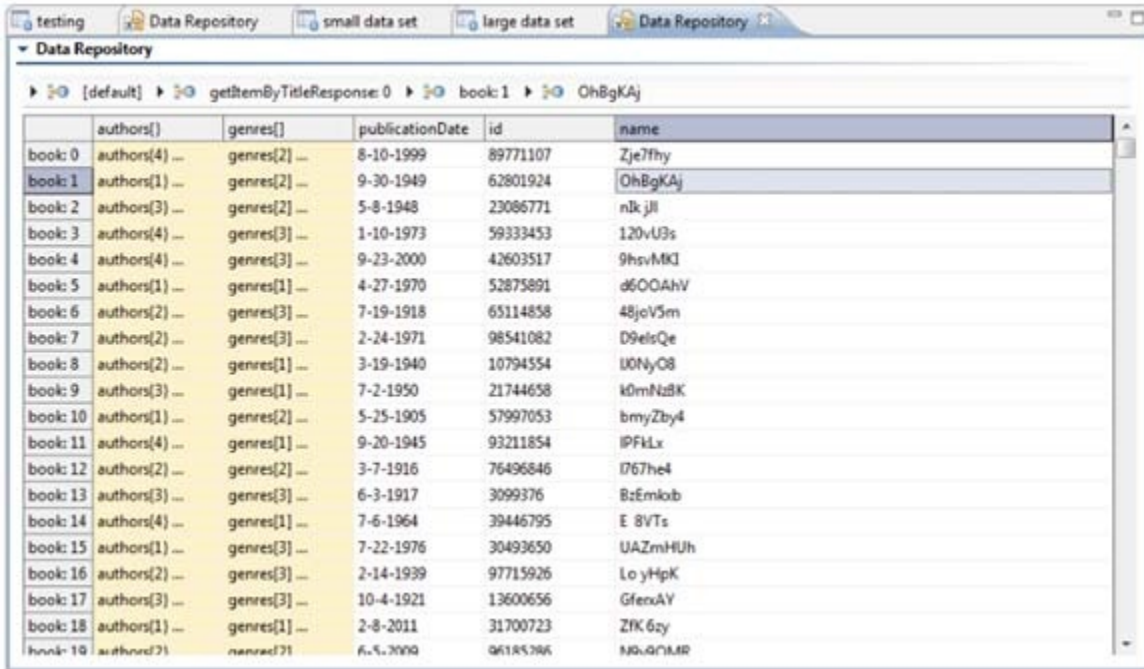
Note that the icon to the left of the repository label indicates the repository status. A full-color icon indicates that the repository is available (i.e., the server hosting it is running). A grayed-out icon indicates that the repository is not available (i.e., the server hosting it is not running).



Icons also alert you if a repository is locked—and if so, by which user. For example, the following screenshot shows a repository locked to the user "cynthia". Locking repositories requires authentication through Continuous Testing Platform (CTP). To show the Data Repositories view, choose **Parasoft> Show View> Data Repositories**.

## Data Editor

The data editor allows you to review and extend the repository structure and contents.



The screenshot shows a software window titled "Data Repository" with a tree view on the left and a data table on the right. The tree view shows a path: [default] > getItemByTitleResponse: 0 > book: 1 > OhBgKAj. The table below contains the following data:

	authors[]	genres[]	publicationDate	id	name
book: 0	authors(4) ...	genres(2) ...	8-10-1999	89771107	Zje7fhy
book: 1	authors(1) ...	genres(2) ...	9-30-1949	62801924	OhBgKAj
book: 2	authors(3) ...	genres(2) ...	5-8-1948	23086771	nIk jll
book: 3	authors(4) ...	genres(3) ...	1-10-1973	59333453	120vU3s
book: 4	authors(4) ...	genres(3) ...	9-23-2000	42603517	9hsvMKI
book: 5	authors(1) ...	genres(1) ...	4-27-1970	52875891	d6OOAhV
book: 6	authors(2) ...	genres(3) ...	7-19-1918	65114858	48joV5m
book: 7	authors(2) ...	genres(3) ...	2-24-1971	98541082	D9eisQe
book: 8	authors(2) ...	genres(1) ...	3-19-1940	10794554	lU0NyO8
book: 9	authors(3) ...	genres(1) ...	7-2-1950	21744658	k0mFHz8K
book: 10	authors(1) ...	genres(2) ...	5-25-1905	57997053	bmyZby4
book: 11	authors(4) ...	genres(1) ...	9-20-1945	93211854	lPFkLx
book: 12	authors(2) ...	genres(2) ...	3-7-1916	76496846	l767he4
book: 13	authors(3) ...	genres(3) ...	6-3-1917	3099376	BzEmlob
book: 14	authors(4) ...	genres(1) ...	7-6-1964	39446795	E 8VTs
book: 15	authors(1) ...	genres(3) ...	7-22-1976	30493650	UAZmHUh
book: 16	authors(2) ...	genres(3) ...	2-14-1939	97715926	Lo yHpK
book: 17	authors(3) ...	genres(3) ...	10-4-1921	13600656	GfexAY
book: 18	authors(1) ...	genres(1) ...	2-8-2011	31700723	ZfK 6zy
book: 19	authors(2) ...	genres(2) ...	6-9-2000	96185296	Ng0G7MBR

## Structure Requirements

The structure of the repository is up to your team. The only requirements are:

- **Message parameterization:** Using a repository data source, you can parameterize messages with sequences or elements with varying numbers of items. Sequences or elements that are of option type or of abstract/concrete schema types cannot be parameterized using repository data.
- **One row column:** The row column is used to determine how SOAtest iterates through a data source. You must have exactly one row column if you will be using this data in SOAtest.