

# User Manual 3.3 Data management system

De Wiki

Aller à : [navigation](#), [rechercher](#)

[User Manual 3.3 Data management system](#)

## Sommaire

- [1 Introduction](#)
  - [1.1 Scope](#)
  - [1.2 Javadoc](#)
  - [1.3 Links](#)
  - [1.4 Useful Documents](#)
  - [1.5 Package Overview](#)
- [2 Features Description](#)
  - [2.1 Default provider](#)
  - [2.2 Adding a provider](#)
  - [2.3 Using the data management system](#)
  - [2.4 Adding new data](#)
- [3 Getting Started](#)
- [4 Contents](#)
  - [4.1 Interfaces](#)
  - [4.2 Classes](#)
- [5 Tutorials](#)
  - [5.1 Tutorial 1](#)
  - [5.2 Tutorial 2](#)
- [6 Tips & Tricks](#)
  - [6.1 Strengths](#)
  - [6.2 Weaknesses](#)

## Introduction

### Scope

The scope of this chapter is to present data management in the OREKIT library. This section presents the three modules of the data management system provided by Orekit :

- how does the data management system of Orekit work?
- how to set it up?
- how to use it?
- how to add data to what already exists?
- what are the pros and cons of this system?

NOTE : this page describes Orekit's data management system assuming it will be used "as is" in PATRIUS- but this is subject to change in later development stages.

## Javadoc

The data objects are available in the package `org.orekit.data` in the Orekit.

### Library      Javadoc

Orekit   [Package org.orekit.data](#)

## Links

[Modèle:SpecialInclusion prefix=\\$theme sub section="Links"/](#)

## Useful Documents

[Modèle:SpecialInclusion prefix=\\$theme sub section="UsefulDocs"/](#)

## Package Overview

The data loading process is organized through three main objects.

The `DataProvider` classes handle data sources. Each one of them has a particular type of source it can browse. The `DirectoryCrawler` performs a bottom-first search in a directory tree. The `ZipJarCrawler` works alike, but inside a compressed file. The `ClassPathCrawler` handles a list of data files and/or compressed files that are in the classpath (it can not search recursively like the `DirectoryCrawler` though). Finally, the `NetworkCrawler` works like the `ClassPathCrawler`, although in its case, it has a list of URLs instead of files. There is no limit to the number of `DataProviders` a program can use at once.

The Providers are listed and put to work through the `DataProvidersManager` singleton. This is the single point of access to the data management system. It contains a list of Providers that are queried every time a user needs data.

The various crawlers provide streams to the `DataLoader`. From these streams, the `DataLoaders` can reconstruct data that was stored in files (either compressed or not), even if some files come from different sources. These streams effectively separate the machine world from the program world, because they hide the former to the latter. Therefore, parsing data from a new format only means creating a loader, and being able to read another kind of file means creating a `DataProvider`. Note that the `DataLoaders` usually serve as a facade for the higher layers of the program.

[Fichier:DataPackageFunctions.png](#)

## Features Description

### Default provider

The data management system can use a system-wide property, `orekit.data.path`, as an entry point for default data. This default data must be file-based (either a file system entry point or a java resource) and either a directory or a zip/jar file. Setting a default provider is not mandatory, and must be done explicitly by :

- setting a value to `orekit.data.path`,

- calling `addDefaultProviders` on the data provider manager.

The Orekit library jar contains data that can be used as default data.

## Adding a provider

[Modèle:SpecialInclusion prefix=\\$theme sub section="Provider"/](#)

## Using the data management system

The data management system main operation is through the `feed` method. This method takes a `DataLoader`, and a regexp string matching the name of files the `DataLoader` is able to process. In this method call :

- the `DataProviders` list is traversed in the priority order.
- the first `DataProvider` providing a file matching the regexp is the one (and only) used to feed the `DataLoader`.

## Adding new data

[Modèle:SpecialInclusion prefix=\\$theme sub section="Adding"/](#)

## Getting Started

[Modèle:SpecialInclusion prefix=\\$theme sub section="GettingStarted"/](#)

## Contents

### Interfaces

The data package includes the following interfaces :

#### Data

Interface	Summary	Javadoc
<b>DataLoader</b>	Interface for loading data files from <code>DataProvider</code> data providers.	<a href="#">...</a>
<b>DataProvider</b>	Interface for providing data files to <code>DataLoader</code> file loaders.	<a href="#">...</a>

### Classes

The data package includes the following classes :

#### Data

Class	Summary	Javadoc
<b>DataProvidersManager</b>	This class is the single point of access for all data loading features.	<a href="#">...</a>
<b>DirectoryCrawler</b>	This class handles data files recursively starting from a root directories.	<a href="#">...</a>
<b>NetworkCrawler</b>	This class handles a list of URLs pointing to data files or zip/jar on the net.	<a href="#">...</a>

## ZipJarCrawler

This class browses all entries in a zip/jar archive in filesystem or in classpath. [...](#)

# Tutorials

## Tutorial 1

[Modèle:SpecialInclusion prefix=\\$theme sub section="Tuto1"/](#)

## Tutorial 2

[Modèle:SpecialInclusion prefix=\\$theme sub section="Tuto2"/](#)

## Tips & Tricks

### Strengths

- Lightweight implementation. The providers never load data, they merely provide streams on demand to the loaders.
- Scalable for using data from several heterogeneous sources.
- Scalable for new data types : the user only needs to create a new `DataLoader` implementation to use a new data type in this system.

### Weaknesses

- The user must be aware the data loading overhead happens any time a `DataLoader` is fed, so the user must manage its loaders so that they are fed only once.
- Several sources for the same type of data cannot be used, since only the last provider added is used to feed data to a loader- unless the user manages the providers list accordingly, knowing one can only add elements or reset the whole list.
- The regexp is the only way to match a data file and a `DataLoader`.
- As of today, the data management system is a thread-hostile singleton : a multithreaded application shares the same providers list for all threads, and it may deadlock on a concurrent access!

Récupérée de

« [http://patrius.cnes.fr/index.php?title=User\\_Manual\\_3.3\\_Data\\_management\\_system&oldid=1464](http://patrius.cnes.fr/index.php?title=User_Manual_3.3_Data_management_system&oldid=1464) »

Catégorie :

- [Pages avec des liens de fichiers brisés](#)

## Menu de navigation

### Outils personnels

- [18.116.14.12](#)
- [Discussion avec cette adresse IP](#)
- [Créer un compte](#)
- [Se connecter](#)

## Espaces de noms

- [Page](#)
- [Discussion](#)

## Variantes

## Affichages

- [Lire](#)
- [Voir le texte source](#)
- [Historique](#)
- [Exporter en PDF](#)

## Plus

## Rechercher

## PATRIUS

- [Welcome](#)

## Evolutions

- [Main differences between V4.15 and V4.14](#)
- [Main differences between V4.14 and V4.13](#)
- [Main differences between V4.13 and V4.12](#)
- [Main differences between V4.12 and V4.11](#)
- [Main differences between V4.11 and V4.10](#)
- [Main differences between V4.10 and V4.9](#)
- [Main differences between V4.9 and V4.8](#)
- [Main differences between V4.8 and V4.7](#)
- [Main differences between V4.7 and V4.6.1](#)
- [Main differences between V4.6.1 and V4.5.1](#)
- [Main differences between V4.5.1 and V4.4](#)
- [Main differences between V4.4 and V4.3](#)
- [Main differences between V4.3 and V4.2](#)
- [Main differences between V4.2 and V4.1.1](#)
- [Main differences between V4.1.1 and V4.1](#)
- [Main differences between V4.1 and V4.0](#)
- [Main differences between V4.0 and V3.4.1](#)

## **User Manual**

- [User Manual 4.15](#)
- [User Manual 4.14](#)
- [User Manual 4.13](#)
- [User Manual 4.12](#)
- [User Manual 4.11](#)
- [User Manual 4.10](#)
- [User Manual 4.9](#)
- [User Manual 4.8](#)
- [User Manual 4.7](#)
- [User Manual 4.6.1](#)
- [User Manual 4.5.1](#)
- [User Manual 4.4](#)
- [User Manual 4.3](#)
- [User Manual 4.2](#)
- [User Manual 4.1](#)
- [User Manual 4.0](#)
- [User Manual 3.4.1](#)
- [User Manual 3.3](#)

## **Tutorials**

- [Tutorials 4.15](#)
- [Tutorials 4.14](#)
- [Tutorials 4.13.5](#)
- [Tutorials 4.12.1](#)
- [Tutorials 4.8.1](#)
- [Tutorials 4.5.1](#)
- [Tutorials 4.4](#)
- [Tutorials 4.1](#)
- [Tutorials 4.0](#)

## **Links**

- [CNES freeware server](#)

## **Navigation**

- [Accueil](#)
- [Modifications récentes](#)
- [Page au hasard](#)
- [Aide](#)

## **Outils**

- [Pages liées](#)

- [Suivi des pages liées](#)
- [Pages spéciales](#)
- [Adresse de cette version](#)
- [Information sur la page](#)
- [Citer cette page](#)

• Dernière modification de cette page le 7 mars 2018 à 07:53.

- [Politique de confidentialité](#)
- [À propos de Wiki](#)
- [Avertissements](#)

- 