

# Table of Contents

Appendix A: List of ItemReaders and ItemWriters .....	1
A.1. Item Readers .....	1
A.2. Item Writers .....	2

# Appendix A: List of ItemReaders and ItemWriters

## A.1. Item Readers

Table 1. Available Item Readers

Item Reader	Description
AbstractItemCountingItemStreamItemReader	Abstract base class that provides basic restart capabilities by counting the number of items returned from an <code>ItemReader</code> .
AggregateItemReader	An <code>ItemReader</code> that delivers a list as its item, storing up objects from the injected <code>ItemReader</code> until they are ready to be packed out as a collection. This <code>ItemReader</code> should mark the beginning and end of records with the constant values in <code>FieldSetMapper</code> <code>AggregateItemReader#BEGIN_RECORD</code> and <code>AggregateItemReader#END_RECORD</code> .
AmqpItemReader	Given a Spring <code>AmqpTemplate</code> , it provides synchronous receive methods. The <code>receiveAndConvert()</code> method lets you receive POJO objects.
FlatFileItemReader	Reads from a flat file. Includes <code>ItemStream</code> and <code>Skippable</code> functionality. See <code>FlatFileItemReader</code> .
HibernateCursorItemReader	Reads from a cursor based on an HQL query. See <code>Cursor-based ItemReaders</code> .
HibernatePagingItemReader	Reads from a paginated HQL query
ItemReaderAdapter	Adapts any class to the <code>ItemReader</code> interface.
JdbcCursorItemReader	Reads from a database cursor via JDBC. See <code>Cursor-based ItemReaders</code> .
JdbcPagingItemReader	Given an SQL statement, pages through the rows, such that large datasets can be read without running out of memory.
JmsItemReader	Given a Spring <code>JmsOperations</code> object and a JMS Destination or destination name to which to send errors, provides items received through the injected <code>JmsOperations#receive()</code> method.
JpaPagingItemReader	Given a JPQL statement, pages through the rows, such that large datasets can be read without running out of memory.
ListItemReader	Provides the items from a list, one at a time.
MongoItemReader	Given a <code>MongoOperations</code> object and a JSON-based MongoDB query, provides items received from the <code>MongoOperations#find()</code> method.

Item Reader	Description
Neo4jItemReader	Given a <code>Neo4jOperations</code> object and the components of a Cypher query, items are returned as the result of the <code>Neo4jOperations.query</code> method.
RepositoryItemReader	Given a Spring Data <code>PagingAndSortingRepository</code> object, a <code>Sort</code> , and the name of method to execute, returns items provided by the Spring Data repository implementation.
StoredProcedureItemReader	Reads from a database cursor resulting from the execution of a database stored procedure. See <code>StoredProcedureItemReader</code>
StaxEventItemReader	Reads via StAX. see <code>StaxEventItemReader</code> .
JsonItemReader	Reads items from a Json document. see <code>JsonItemReader</code> .

## A.2. Item Writers

Table 2. Available Item Writers

Item Writer	Description
AbstractItemStreamItemWriter	Abstract base class that combines the <code>ItemStream</code> and <code>ItemWriter</code> interfaces.
AmqpItemWriter	Given a Spring <code>AmqpTemplate</code> , it provides for a synchronous <code>send</code> method. The <code>convertAndSend(Object)</code> method lets you send POJO objects.
CompositeItemWriter	Passes an item to the <code>write</code> method of each in an injected <code>List</code> of <code>ItemWriter</code> objects.
FlatFileItemWriter	Writes to a flat file. Includes <code>ItemStream</code> and Skippable functionality. See <code>FlatFileItemWriter</code> .
GemfireItemWriter	Using a <code>GemfireOperations</code> object, items are either written or removed from the Gemfire instance based on the configuration of the delete flag.
HibernateItemWriter	This item writer is Hibernate-session aware and handles some transaction-related work that a non-"hibernate-aware" item writer would not need to know about and then delegates to another item writer to do the actual writing.
ItemWriterAdapter	Adapts any class to the <code>ItemWriter</code> interface.
JdbcBatchItemWriter	Uses batching features from a <code>PreparedStatement</code> , if available, and can take rudimentary steps to locate a failure during a <code>flush</code> .

Item Writer	Description
JmsItemWriter	Using a <code>JmsOperations</code> object, items are written to the default queue through the <code>JmsOperations#convertAndSend()</code> method.
JpaItemWriter	This item writer is JPA EntityManager-aware and handles some transaction-related work that a non-"JPA-aware" <code>ItemWriter</code> would not need to know about and then delegates to another writer to do the actual writing.
MimeMessageItemWriter	Using Spring's <code>JavaMailSender</code> , items of type <code>MimeMessage</code> are sent as mail messages.
MongoItemWriter	Given a <code>MongoOperations</code> object, items are written through the <code>MongoOperations.save(Object)</code> method. The actual write is delayed until the last possible moment before the transaction commits.
Neo4jItemWriter	Given a <code>Neo4jOperations</code> object, items are persisted through the <code>save(Object)</code> method or deleted through the <code>delete(Object)</code> per the <code>ItemWriter</code> 's configuration
PropertyExtractingDelegatingItemWriter	Extends <code>AbstractMethodInvokingDelegator</code> creating arguments on the fly. Arguments are created by retrieving the values from the fields in the item to be processed (through a <code>SpringBeanWrapper</code> ), based on an injected array of field names.
RepositoryItemWriter	Given a Spring Data <code>CrudRepository</code> implementation, items are saved through the method specified in the configuration.
StaxEventItemWriter	Uses a <code>Marshaller</code> implementation to convert each item to XML and then writes it to an XML file using StAX.
JsonFileItemWriter	Uses a <code>JsonObjectMarshaller</code> implementation to convert each item to Json and then writes it to an Json file.