

0-Recommended Maximum File Sizes and Configuration

Based on analysis by the Performance Task Force during the Juniper development cycle, these are the recommended maximum file sizes and key configuration settings for Data Import as of the Iris Hotfix 3 and Juniper release. Please work with your systems office or hosting to provider to ensure these configurations are in place. After more performance, reliability, and stability work during Kiwi, these recommendations will be re-evaluated.

For additional information, please see the following:

- [Folijet - Morning Glory Snapshot Performance testing](#)
- [MARC Bibs: Performance Rancher testing \(Lotus\)](#)
- [MARC Bibs: Performance Testing Data Import \(Juniper/Kiwi\)](#)
- [MARC Bibs: Performance Testing Data Import \(Iris\)](#)
- [MARC Bibs: Profiles used for PTF Testing](#)
- [MARC Bibs: PTF Test results](#)

Maximum File Size (Lotus) no background activity, no concurrent import jobs

- **CREATE** (SRS MARC, Instances, Holdings, Items): **100,000 MARC Bib records** ([details](#))
- **UPDATE** (SRS MARC, Instances, Holdings, Items): **50,000 MARC Bib records**
- Still testing Data Import Lotus performance with background activity and concurrent import jobs

Import Statistics, with recommended values no background activity, no concurrent import jobs

MARC Records	Lotus				Juniper and Kiwi	
	1 instance		2 instances		2 instances	
	CREATE Duration	UPDATE Duration	CREATE Duration	UPDATE Duration	CREATE Duration	UPDATE Duration
5000	13min	16min	8min	13min	15 minutes	20-30 minutes (depending on complexity of profile)
10,000	31min	40min	19min	25min		
30,000	1h 31min	1h 4 min	45min	1h 36min		
50,000	1h 34min	2h 17min	1h 21min	1h 51min	2h 30min+	11+ hours
100,000	3h 21min		3h 10min	4h 10min		

Maximum File Sizes (Juniper and Kiwi)

- **CREATE** Import (SRS MARC, Instances, Holdings, Items): **50,000 MARC records max**
- **UPDATE** Import: **5,000 MARC records max**

Import Statistics, with background activity

- With 5-users check-in/out background activities
- And concurrent imports by different tenants
 - Check-in/out time increases by 50%-100% depending on number of concurrent users.
 - Data import takes 2x longer to complete.

MARC Records	CREATE Duration	UPDATE Duration
1,000	10 minutes	10 minutes
5,000	30 minutes	40-60 minutes (depending on complexity of profile)
25,000	2-3 hours	8+ hours
50,000	5+ hours	22+ hours

Import Statistics, no background activity, no concurrent import jobs

MARC Records	CREATE Duration	UPDATE Duration
1,000	5 minutes	5 minutes
5,000	15 minutes	20-30 minutes (depending on complexity of profile)
25,000	60-80 minutes	4+ hours
50,000	150+ minutes	11+ hours

Note: For Lotus 500,000 Create Duration -15h 37min ([details](#))

Key Settings and Configurations

More information on configuring modules involved in Data Import process can be found at this [link](#).

- **Kafka (MSK):**
 - auto.create.topics.enable = true
 - log.retention.minutes = 70-300
 - Broker's disk space: 300 GB
 - 4 brokers, replication factor = 3, DI topics partition = 2
 - Version 2.7 is 30% faster than version 1.6
- **mod-inventory:**
 - inventory.kafka.DataImportConsumerVerticle.instancesNumber=X (the default value is 3 and if number of kafka partitions greater than 3 and scaling not enabled - in this case we recommend to increase value of this property)
 - inventory.kafka.MarcBibInstanceHridSetConsumerVerticle.instancesNumber=X (the default value is 3 and if number of kafka partitions greater than 3 and scaling not enabled - in this case we recommend to increase value of this property)
 - kafka.consumer.max.poll.records=10
 - Memory: 2 GB
- **mod-data-import (applicable only for releases prior to Kiwi):**
 - file.processing.buffer.chunk.size=5 (for Update import of 5,000 records, set to 1 in case import of larger files is expected)

Key Improvements Delivered for Morning Glory:

Key Improvements Delivered for Kiwi/Lotus:

- Performance
 - Improve speed and number of records for CREATEs and UPDATEs
 - Reduce Kafka message size: 6 Data Import topics messages >200KB/ea
 - Use less CPU for (de)serialization
 - Significantly improve speed of loading UI Landing page
 - Improve and optimize slow DB queries
- Remove events_cache topic and replace it by DB deduplication solutions (causes spikes in mod-inventory and the brokers, leads to instability and unpredictable outcomes)
- Improve resiliency and error handling to prevent import job from getting stuck

Key Improvements Delivered as of Iris Hotfix 3 and Juniper:

- No more accidental creation of duplicate records
- Ability to consistently make repeated updates on existing records
- CPU usage of multiple instances of the Data Import landing page now consumes 10% CPU, instead of maxing out at 100%
- When idle, mod-source-record-manager consumes 50% CPU instead of 90%
- Vertical scaling improves performance, especially with Iris HF2 Data Import modules