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:

> **Performance Testing Data Import (Iris)**
>
> **Profiles used for PTF Testing**
>
> **PTF Test results**
>
> **Performance Testing Data Import (Juniper/Kiwi)**

Maximum File Sizes (Juniper)

- **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

Key Settings and Configurations

- **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 = 1
    - Version 2.7 is 30% faster than version 1.6
- **mod-inventory:**
    - inventory.kafka.DataImportConsumerVerticle.instancesNumber=10
    - inventory.kafka.MarcBibInstanceHridSetConsumerVerticle.instancesNumber=10
    - 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)

More information on configuring modules involved in Data Import process can be found by the link.

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

Additional work (planned for Kiwi and perhaps 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
- Remove events_cache topic (causes spikes in mod-inventory and the brokers, leads to instability and unpredictable outcomes)
- Resiliency when modules crash (module crashes may leave the job stuck or finished with not all records created/updated)

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:

Performance Testing Data Import (Iris)

Profiles used for PTF Testing

PTF Test results

Performance Testing Data Import (Juniper/Kiwi)

Maximum File Sizes (Juniper)

- **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

Key Settings and Configurations

- **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 = 1
  - Version 2.7 is 30% faster than version 1.6
- **mod-inventory:**
  - inventory.kafka.DataImportConsumerVerticle.instancesNumber=10
  - inventory.kafka.MarcBibInstanceHridSetConsumerVerticle.instancesNumber=10
  - 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

More information on configuring modules involved in Data Import process can be found by the link.

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

Additional work (planned for Kiwi and perhaps 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
- Remove events_cache topic (causes spikes in mod-inventory and the brokers, leads to instability and unpredictable outcomes)
- Resiliency when modules crash (module crashes may leave the job stuck or finished with not all records created/updated)