insightsoftware

Enterprise Integration Guide

LeaseAccelerator

Version 26.2

Document Information

Notices

Copyright

Disclaimer

This guide is designed to help you to use the LeaseAccelerator applications effectively and efficiently. All data shown in graphics are provided as examples only. The example companies and calculations herein are fictitious. No association with any real company or organization is intended or should be inferred.

Contents

About LeaseAccelerator Integrations

LeaseAccelerator is the leading enterprise lease accounting solution for equipment and real estate.

LeaseAccelerator is a cloud-based SaaS application that integrates with on-premise and cloud applications to enable straight-through processing between critical financial applications. LeaseAccelerator is hosted at Amazon Web Services, the largest global web hosting service provider, with an additional provider for backup and disaster recovery protection.

LeaseAccelerator ensures that all your leases comply with the latest lease accounting standards. It can quickly generate all the debits and credits you need for your general ledger and calculates the corresponding due payments in addition to providing a library of reports and dashboards to monitor and control your leasing business. For equipment leases, LeaseAccelerator will account for aircrafts, trucks, computers, photocopiers, and other types of equipment leases. For real estate leases, it will account for all your office buildings, retail stores, distribution centers, and healthcare facilities.

This guide covers integration requirements and common business cases that are typically required for process and data integration between LeaseAccelerator and other systems.

Disclaimer:

LeaseAccelerator exerts every effort to assure that all data exported are correct and accurate and all data imported are correctly handled and ingested in the LeaseAccelerator database. However, clients elect to use LeaseAccelerator integration tools at their own discretion and under their full responsibility without any liability toward LeaseAccelerator. Clients must set the necessary procedures and controls to review, examine and verify data exchanged between LeaseAccelerator and external systems.

Also, please be aware that Beta environment is designated for testing purposes. We never recommend extracting data from a test environment to use in your ERP or other production environment systems.

Integration with ERP and External Systems

Typically, LeaseAccelerator generates financial entries for your accounting system or ERP. LeaseAccelerator has multiple integration methods to export transactions in different forms so they can be uploaded into your accounting system or ERP. LeaseAccelerator can also receive data from external systems such as master (reference) data and currency exchange rates.

LeaseAccelerator offers two different methods to exchange data (inbound and outbound) with external systems such as ERPs, real estate administration systems, fixed asset systems, and any other system. Those methods include:

1 – Application Programming Interface: RESTful APIs.

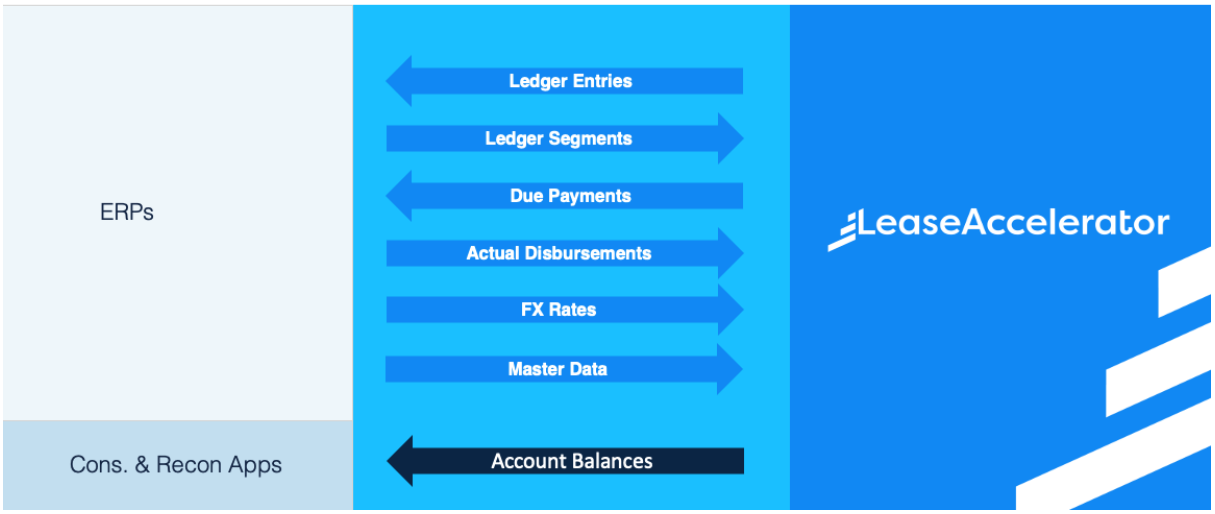
2 – File-based Integration through client FTP /SFTP servers.

Any of the above-mentioned integration methods can be used for:

- Synchronizing reference data such as lessee, project, property tax authority, address, business unit, entity, funder, geo, company, and contact information
- Importing leased assets and deals as well as data for subledger entries such as disbursements and payment adjustments
- Recording events or intentions for future events such as lease terminations, renewals, and buyouts
- Generating ledger exports and asset-level detail reports
- Generating due payments to ERP Accounts Payable and importing actual disbursements
- Importing currency exchange rates
- Searching for leased assets, contacts, and schedules
- Provisioning and revoking user privileges for Single Sign-On

This guide covers the following integrations:

Enterprise Integration



Some companies use vertical solutions to manage the operation, maintenance, inspection, and utilization of leased real estate, fleet or equipment. In such cases, LeaseAccelerator can exchange information with these systems. Please refer to the Portfolio Integration Guide for details.

Portfolio Integration

Integration Strategies

LeaseAccelerator has a proprietary robust integration methodology that ensures delivered integrations are business-driven, reliable, secured, and sustainable.

Business-driven

To maintain flexibility and adapt integrations to different business requirements, LeaseAccelerator provides a set of tools and interfaces that can be easily configured to suit specific business needs.

Accordingly, each client — after reading this guide — should be able to select the business workflows, use cases, and mapping that meet the company's specific needs and standards.

Each client's requirements are translated into an "Integration Business Solution Architecture" which is then translated into the "Technical Solution Architecture" that uses the rich variety of tools and application interfaces that LeaseAccelerator offers.

System-agnostic

Every client has different integration requirements and limitations. LeaseAccelerator offers multiple interfacing methods that serve different business and technical needs and support integrations regardless of the system.

Agile and updated

LeaseAccelerator's Engineering team is continuously improving the application's functional and technical capabilities, including the utilization of the latest interfacing techniques and standards.

Sustainable

A prime objective for LeaseAccelerator is to make sure that interfaces are reliable and will function properly in the present and the future. LeaseAccelerator maintains partnerships and alliances with major ERP vendors, and monitors upgrades and updates, particularly to integration interfaces.

Secure

LeaseAccelerator integration methods ensure maximum security and data protection.

Access to LeaseAccelerator is secured by SAML 2 (Security Assertion Markup Language version 2.0), allowing for instantiation of a secure session that can then be used for multiple requests. To support this secure access, the client's systems must be integrated with LeaseAccelerator's single sign-on (SSO) solution. Assuming you already have a SAML 2-compliant identity provider configured, this is generally a simple matter of exchanging SAML 2 metadata, consisting of certificate and configuration information.

LeaseAccelerator's API uses industry standard HTTPS (secure hypertext transfer protocol) with SAML 2 authentication for data transfer.

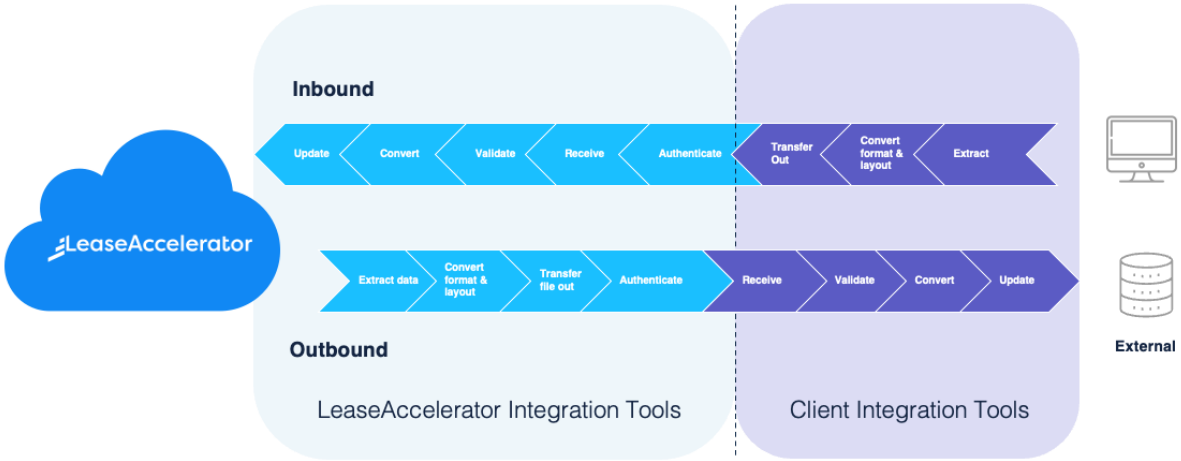
LeaseAccelerator's file-based transfer is secured over SFTP (secure file transfer protocol). Exported files can be encrypted using PgP if requested.

Mail-based features are secured by requiring TLS (transport layer security) between the client and LeaseAccelerator email domains.

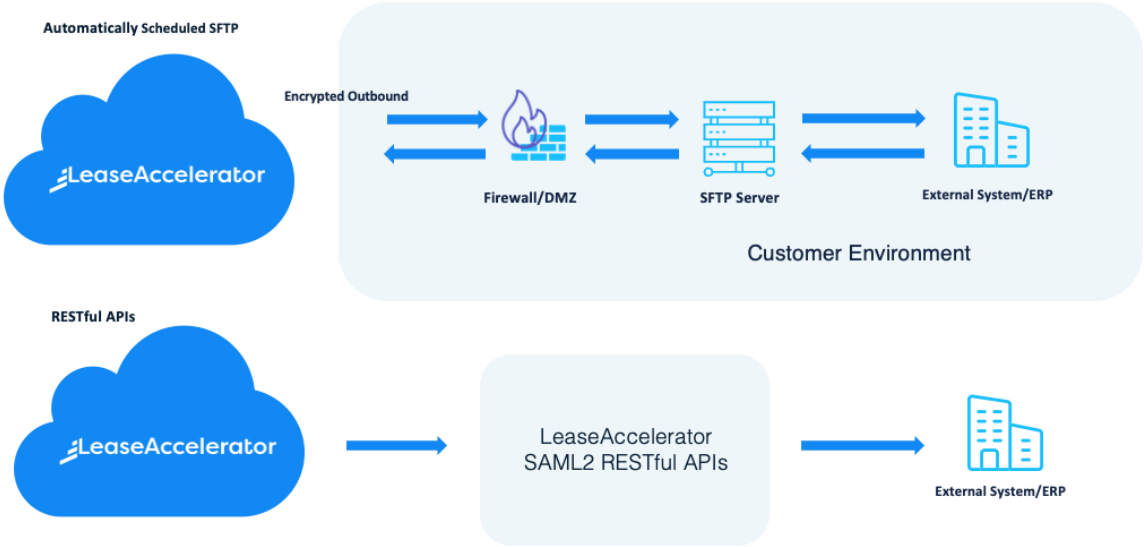
All exported files though a file-based method can be encrypted using PgP and all API requests are protected by authentication keys.

Integration Methods

LeaseAccelerator offers API and file-based integrations, both of which mitigate security risks.



Integration Methods: Point to Point



Application Programming Interfaces (APIs)

For inbound and outbound integration, LeaseAccelerator offers a full library of well-documented and secured RESTful APIs to interface with ERPs, real estate administration systems, fleet management systems, fixed asset systems, and any other system that can call RESTful APIs and process XML.

LeaseAccelerator Restful APIs can be used for:

- Synchronizing reference data such as Lessee, project, property tax authority, address, business unit, entity, funder, geo, company, and contact information

- Importing leased assets and deals as well as data for subledger entries such as disbursements and payment adjustments
- Recording events or intentions for future events such as lease terminations, renewals, and buyouts
- Generating ledger exports and asset-level detail reports
- Generating due payments to ERP Accounts Payable and importing actual disbursements
- Importing currency exchange rates
- Searching for leased assets, contacts, and schedules
- Provisioning and revoking user privileges for Single Sign-On

File-based

In addition to APIs, LeaseAccelerator supports simple file transfer and file exchange.

Outbound file exports:

LeaseAccelerator generates and exports files in multiple formats: XML, TSV, CSV, Pipe delimited, fixed-width or XLSX. While for imports, LeaseAccelerator can import files in csv, xml and XLSX (Microsoft Excel) formats.

The Reporting Engine supports automatic transfer of generated reports via SFTP (secure file transfer protocol) based on a user defined schedule, allowing reporting to be leveraged as a data export mechanism.

For exports (LeaseAccelerator outbound); LeaseAccelerator users can export files manually at any time or can schedule files to be automatically generated and transferred to a designated folder on a client-

provided SFTP folder (see section FTP SETUP FOR OUTBOUND FILE TRANSFER) and/or e- mail at specific frequency (daily, weekly, monthly, etc.). The client then uploads these files into the destination ERP/ application.

To ensure security and protection, LeaseAccelerator can encrypt the exported files to SFTP folder using PgP.

LeaseAccelerator Inbound file Transfer import:

LeaseAccelerator can import files in csv, xml and XLSX (Microsoft Excel) formats.

For imports (inbound), there is a one-time configuration where LeaseAccelerator file watcher can be configured by the LeaseAccelerator team to monitor a specific folder on the client FTP server.

Once files with specific, agreed upon name, format and layout are placed in this folder, LeaseAccelerator processes the file by validating and importing the data into LeaseAccelerator then sending a notice of failure or success via email.

Integration project planning

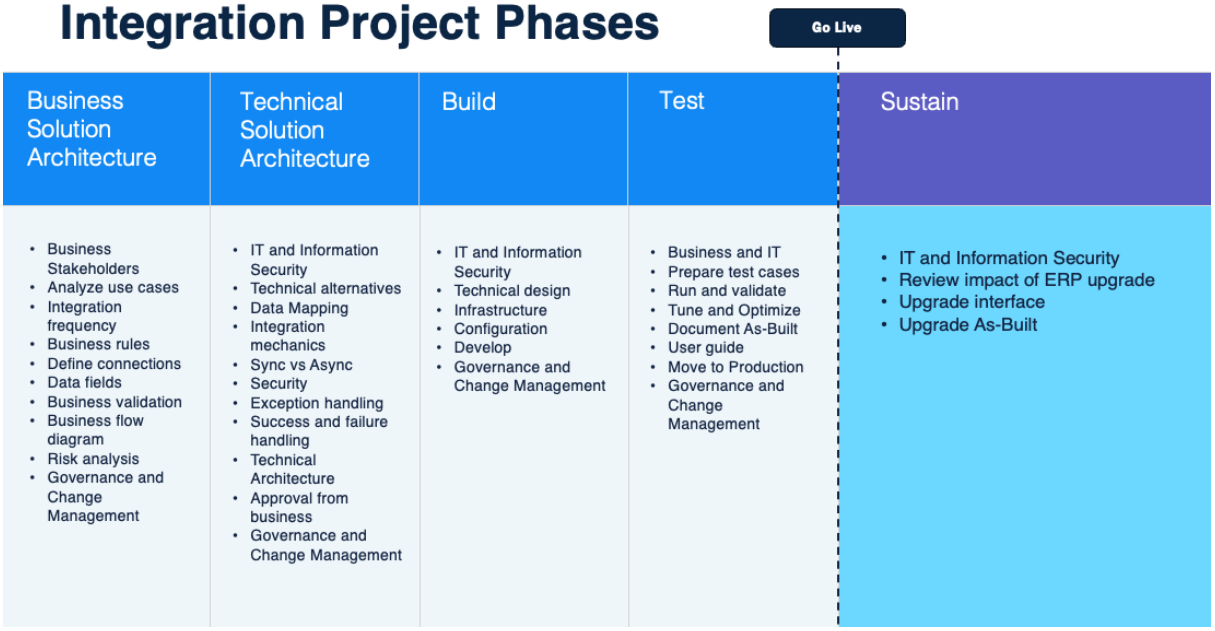
Professional Services team

The integration project is led by LeaseAccelerator's full-time Professional Services team which has extensive experience in implementing LeaseAccelerator, including integration to major ERPs and asset management systems.

LeaseAccelerator's integration professionals manage the integration project and lead the development of the business and technical solution architectures and designs for integration. Based on the solution selected and due to the variety of ERPs and external systems, the actual build of the interfaces may require the services of external implementation consultants or the client's IT resources.

Integration project phases

The LeaseAccelerator integration project is completed in five stages that are led by the Professional Services team at LeaseAccelerator.



Business solution architecture

The main purpose of any integration is to enable systems to exchange information and allow a business process to span across two systems.

Because every client has different business and technical requirements and limitations, LeaseAccelerator has a variety of methods and tools to support all forms of integration, including real time, asynchronous, file-based, and bulk upload.

The first phase of any integration is to discuss the business requirements with stakeholders.

To determine the requirements — and ensure that they meet the client's long-term business needs — LeaseAccelerator considers the post-integration process and analyzes all scenarios to ensure that they account for all use cases.

Business stakeholders are engaged to review and analyze the following:

- Use cases
- Data exchange frequency
- Business rules
- Connections
- Data fields
- Business validation scenarios and exception handling
- Business flow
- Risks

During this first phase, business stakeholders begin to build test cases covering all possible scenarios.

Technical solution architecture

Once the business solution is approved, the integration technical professionals from IT analyze it and draft a technical solution to satisfy the requirements and factor in technology, security, and sustainability.

The IT and InfoSec teams analyze the following:

- Technical alternatives of integration methods
- Data mapping
- Internal mechanics
- Sync vs. async data exchange
- Authentication
- Security
- Sustainability
- Handling for successful data exchanges, failures, and exceptions
- Potential ERP upgrade effects on proposed integration design

If the technical architecture would affect any of the business requirements, the technical integration team returns to the business stakeholders to agree on changes and approve the final solution architecture.

During this phase, the technical stakeholders begin to build test cases, covering all possible scenarios. Often, technical teams build their cases on top of the business test cases.

Build

Depending on the solution architecture and design, a build team may be responsible for the following:

- Configuring the applications involved in the integrations
- Providing infrastructure support, including assembly of components and middleware
- Customizations – if any
- Documenting the integration

Test

Once the build phase is complete in a test environment, the business and technical teams run their test cases, validate and register results, and connect with the build team for any support needed.

Once all test cases have been executed successfully, the build team tunes and optimizes the integration.

After documentation of the integration as built is complete, the technical team is ready to move the integration to the production environment.

Sustain

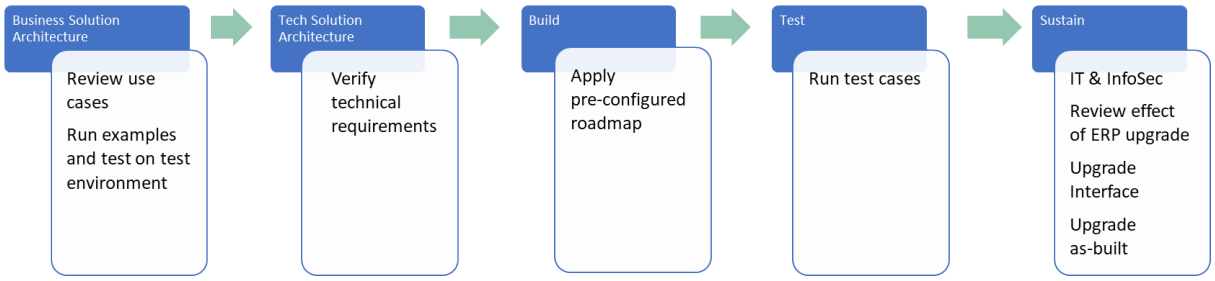
To ensure that the integration is sustainable for the long term, the integration team should proactively consider any potential impacts, which may include the following:

- Upgrade of one of the integrated systems
- Upgrade of an operating system or middleware
- Changes in security policy or authentication technologies
- Changes in business requirements
- Changes in data load volumes handled by the interfaces
- Changes in performance
- Frequency of failure and data corruption
- Security vulnerabilities

The Fast Track approach

This guide explains standard and best practice process flows, use cases, exceptions, and data mapping, based on tested and optimized solutions. Clients who follow the contents, approaches, and

mapping can go into the fast-track approach explained in the chart below which saves up to 60% of the standard time needed for integration.



Fast Track integration activities

Phase 1: Solution Architecture

#	Task name	Typical LeaseAccelerator effort hrs.	Lease Accelerator role	Client role
1	**Solution Architecture**			
1.1	**Business Solution Architecture**			
1.1.1	Review relevant Integration Guide	2	C	R
1.1.2	Discuss integration business requirements	2	C	R
1.1.3	Workshop to review use cases, end-to-end processes, frequency	3	C	R
1.1.4	Discuss and agree on data mapping	2	R	R
1.1.5	Finalize business solution architecture	2	C	R
1.2	**Technical Solution Architecture**			
1.2.1	Workshop for integration methods and alternatives	1	C	R
1.2.2	Workshop to agree on integration mechanics, security, and encryption	2	C	R
1.2.3	Agreement on TSA & mapping	1	R	R
1.3	Update business solution architecture	4	R	C
1.4	Client's final agreement on architecture	1	C	R
	Total	**20**		

Phase 2: Build and Test

#	Task name	Typical LeaseAccelerator effort hrs.	LeaseAccelerator role	Client role
2	**Build and Test**			
2.1	**Build**			
2.1.1	Client to prepare business test cases		C	R
2.1.2	Client to prepare technical test cases		C	R
2.1.3	Review test cases	3	R	R
2.1.4	Discuss and finalize test cases	3	R	R
2.1.5	LeaseAccelerator to enable integration methods	2	R	C
2.1.6	LeaseAccelerator to apply light touches to integration files	3	R	C
2.1.7	Verify data mapping	2	R	C
2.1.8	Client to configure middleware		C	R
	Client to develop custom programs and prepare data sources / target		C	R
2.1.9	Test Integration method / infrastructure	2	R	R
2.2	**Testing**			
2.2.1	Client to prepare test environment		C	R
2.2.2	Configure test Environment	3	C	R
2.2.3	Conduct testing	4	C	R
2.2.4	Refine integration and handle exceptions	3	R	C
2.2.5	Re-run test scenarios	2	C	R
2.2.6	Handle comments and findings	4	R	C
2.2.7	Finalize integration manual	4	R	C
2.2.8	Client to sign-off integration in test environment	1	C	R
3	**Migrate to production**	4	R	R
	Total	**40**		

General Ledger Integration

Direct integration of LeaseAccelerator with the general ledger eliminates the need for manual extraction and upload of the debits and credits from the leasing subledger. Direct integration ensures that the general ledger is kept in sync, minimizing the risk of journal entries being unintentionally excluded from reporting results.

Business architecture

LeaseAccelerator is the subledger for your lease accounting. To support the monthly close, a consolidated set of journal entries from the leasing subledger (LeaseAccelerator) must be posted into the ERP general ledger.

LeaseAccelerator maintains all information about your leasing contracts (sometimes named schedules or deals) including all necessary asset lease details, accounting information, accounting classification, payment schedule, end-of-term actions, and lessors' information.

LeaseAccelerator users can define as many ledgers (sets of books) as needed. Each ledger can support up to 11 *segments* (dimensions such as business units, cost centers, profit centers, etc.). to match ERP set(s) of books. (Please refer to LeaseAccelerator configuration and usage guides for details).

When lease contracts (schedules / deals) are *booked* into LeaseAccelerator, users link each deal to one or more ledgers. Based on this information, LeaseAccelerator generates the general journal entries (named as ledger entries) that must be posted to the ERP general ledger module. For each journal entry, LeaseAccelerator transfers details such as account number, account name, transaction amount, ledger date, segments, transactional currency, and debit or credit designation.

LeaseAccelerator generates ledger entries in *transactional* currencies. Ledger entries include "FX Conversion Date," "FX Rate Type," and "Currency," so when posted in the ERP, the General Ledger module can apply the proper conversion rate for the transaction currency at the specified conversion date.

LeaseAccelerator also provides different "Key identifiers" for each entry line or group. Once ledger entries are successfully posted into the ERP General Ledger module, posting acknowledgment must be sent back to LeaseAccelerator using the key identifiers so that LeaseAccelerator can mark these entries as "Posted" as explained in detail below.

LeaseAccelerator can interact with multiple general ledgers running on different ERP instances from different vendors, including a mix of SAP, Oracle, and other applications.

Pre-posting and post-review good practice

The ledger export report can be generated manually whenever needed and can also be scheduled to be automatically generated and sent by email in a chosen format at specific days (relative to the fiscal period start and end dates).

It is a good practice to apply the following processes:

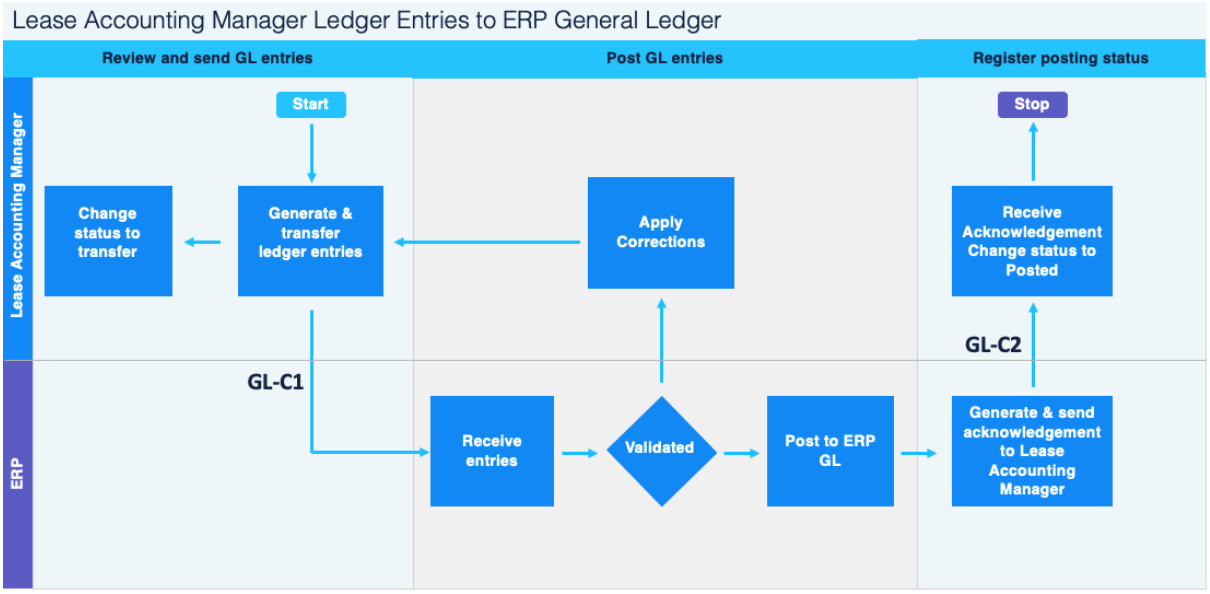
- Close the fiscal period in LeaseAccelerator.
- Once closed (at least 2 days before the planned integration day where ledger entries shall be sent to the ERP GL for integration), generate ledger export (can be scheduled or manually through the UI) and send it by email for revision by the lease accountant.
- The accountant will have two days to review and, if necessary, re-open the period in LeaseAccelerator and make any corrections needed then re-close the period.
- On the specified day, the integration officer will execute the integration and post entries into the ERP GL.
- The integration officer shall extract the ERP posting reference IDs (PostingId, PostingEntryId and LedgerEntryLineId) and complete the posting acknowledgment roundtrip back to LeaseAccelerator and acknowledge the functional users.
- The user shall manually generate the Ledger Export report again and choose to exclude all “Posted” entries, this would be an exception report showing all non-posted entries. The report should come up empty.

General rules:

- Only generate Ledger entries when the fiscal period in LeaseAccelerator is **closed**. If a user generates a ledger export file while the fiscal period is open, another user may change or add data to lease contracts that may trigger LeaseAccelerator to generate different sets of ledger entries.
- Once ledger entries are generated, the period should **remain closed** until the posting acknowledgment roundtrip is completed. Unless the generated file is trashed and not posted into the ERP GL.

Integration process flow

Typically, immediately after closing the fiscal period in LeaseAccelerator and before month-end closing in the ERP’s General Ledger, leasing journal entries need to be transferred from LeaseAccelerator to the general ledger where they should be posted.



Process #	Process Step Description	Phase
P100	Start	Review and send GL entries
P200	Generate & Transfer Ledger Entries	Review and send GL entries
P300	Receive Entries	Post GL Entries
P400	Validated?	Post GL Entries
P500	Post to ERP GL	Post GL Entries
P600	Generate & Send Acknowledgment to LeaseAccelerator	Post GL Entries
P700	Receive Ack Change status to “Posted”	Resend GL Entries
P800	Apply Corrections	Resend GL Entries

P100: Start

The general ledger integration process must be triggered immediately before month-end closing in the ERP and must be executed for each ledger. The client executes the process for the primary and the secondary ledgers.

The fiscal month in LeaseAccelerator must be closed first to prevent any amendments or changes to LeaseAccelerator before posting the entries into the ERP and sending back the posting acknowledgment to LeaseAccelerator.

P200: Generate & transfer ledger entries

LeaseAccelerator provides a detailed ledger export report that shows the entries that will be sent to the general ledger. Regardless of the integration method selected, it is advised that accounting users should generate this report and review these entries before executing the integration.

Each set of books requires a separate general ledger integration exchange.

LeaseAccelerator initially marks the ledger entries as “New”. And, by default, the report shows entries marked as “New” or “Transferred” only.

Users can define a specific schedule whereby LeaseAccelerator automatically generates and sends the report in the specified format to designate users either by email or by SFTP to a specific destination.

Users can activate GL-C1 integration connection to transfer the entries as follows:

Method	Process
RESTful API	ERP integration adapter initiates an API call to LeaseAccelerator to generate a ledger export message. LeaseAccelerator generates an XML response containing the ledger entries for the requested fiscal period.
File-based	Choose the Transfer option whereby LeaseAccelerator creates the ledger export file and posts it to the designated client FTP /server folder.

Upon successful transfer, LeaseAccelerator changes the status of the transferred ledger entries within the LeaseAccelerator internal database from “New” to “Transferred”.

LeaseAccelerator stores the transmitted ledger entries with posting referenceIDs, time/date of request, request parameters, and user requesting the transfer.

P300: Receive entries (GL-C1)

The ERP receives the ledger entries from LeaseAccelerator depending on the integration method as follows:

Method	Process
RESTful API	ERP middleware receives the API response payload (ledger entries), validates it, and ingests the ledger entries into the ERP.
File-based	In the agreed upon file exchange destination folder on the client-provided SFTP server, a mechanism detects the new file and initiates ingest of the file so that journal entries are staged in the general ledger as unposted entries.

P400: Validated?

The ERP integration officer runs tools to validate the entries in the ledger export file.

The validation tools may discover two types of errors that will prevent posting these entries into the ERP general ledger as follows:

- Soft errors such as ledger codes, currency code, or accounts are invalid, or a null value was received for a non-null field.
- Hard errors, which usually result from file corruption due to transmission or disk I/O errors.

If an invalid ledger coding (either segment value or combination of segment values) is detected, the entire batch should be rejected and remanded to accounting for correction. Partial ingest is not recommended.

If there are errors, proceed back to step P200.

P500: Post to general ledger

Once the ledger entries file is validated, the ERP technical integration officer posts the ledger entries into the general ledger and records the success process in the integration log. The ERP generates a posting ID for each entry posted.

P600: Generate & send acknowledgement to LeaseAccelerator (GL-C2)

Journal entries that have been accepted into the general ledger are posted and should not be re-sent by LeaseAccelerator. The general ledger needs to advise LeaseAccelerator of these posted journal entries.

Method	Process	Next Step
RESTful API	As a trigger or as a nightly process, the ERP recognizes posting of ledger entries that carry a LeaseAccelerator journal posting reference ID. The ERP generates an XML payload containing the set of LeaseAccelerator journal posting reference IDs posted along with the associated ERP posting ID for each. The ERP integration adapter initiates an API call to LeaseAccelerator to capture document ID with the generated payload.	P700
File-based	The ERP technical integration officer runs a procedure that generates a file containing the set of LeaseAccelerator journal posting reference IDs posted along with the associated ERP posting ID for each entry. The file is staged in the configured SFTP location.	P700

P700: Receive ack change status to “posted” (GL-C2)

Method	Process	Next Step
RESTful API	LeaseAccelerator validates and imports the payload (acknowledgements) from the ERP. LeaseAccelerator acknowledges the request with an OK response.	STOP
File-based	LeaseAccelerator’s ingest task fires on a configured periodicity (as often as every ten minutes; as infrequently as weekly) and checks the configured integration point for new files. Recognizing that a new, unprocessed file has arrived, LeaseAccelerator validates and imports the file. LeaseAccelerator removes the detected file from the SFTP location.	STOP

For each referenced LeaseAccelerator journal posting reference ID, LeaseAccelerator marks the associated ledger entries as “Posted” and tags them with the posting IDs from the ERP.

LeaseAccelerator sends an email notification to the designated recipients that the file has been successfully received and processed.

The process ends here.

P800: Apply corrections

Depending on the errors detected, users would take necessary actions to correct those errors and regenerate the file.

Once corrections have been made per step P200, users should re-close the fiscal period in LeaseAccelerator then regenerate the ledger export file.

This time, the user must choose entries marked as “New” and “Transferred” since the transferred entries were rejected.

Understanding Posting Acknowledgement

LeaseAccelerator should be acknowledged about the ledger entries which have been successfully posted in the ERP General Ledger so that LeaseAccelerator would:

1. Exclude posted entries from successive ledger exports for the same period, preventing exporting duplicate entries to the ERP.

2. Lock the posted entries so that if the fiscal period was reopened and users applied changes to schedules that require the accounting engine to change the ledger entries, LeaseAccelerator accounting engine would generate new incremental and reverse entries without doing any changes to these locked (posted) entries.

LeaseAccelerator expects to receive acknowledgment file to be sent back carrying a list of 2 fields:

1. Reference IDs: LeaseAccelerator accepts any of the abovementioned keys (PostingId, PostingEntryId or LedgerEntryLineId)

2. External Document ID: Any text (cannot be empty) that should include posting reference in the ERP GL to enable cross-system tracing.

You can send back LedgerEntryLineId(s) or the PostingEntryId(s) or the PostingId(s) as the reference key to LeaseAccelerator to acknowledge posting of the corresponding entries.

Using LEDGERENTRYLINEID vs. POSTINGID vs. POSTINGENTRYID

LeaseAccelerator generates ledger entries, and these entries should be posted in the ERP General Ledger module.

To maintain synchronization between LeaseAccelerator and the ERP and prevent LeaseAccelerator from changing or deleting entries that were already posted in the ERP GL, LeaseAccelerator needs a “return receipt” or “Posting Acknowledgment”.

To enable this posting acknowledgment LeaseAccelerator Ledger entries, include one of three different key identifiers: LedgerEntryLineId, PostingEntryId and PostingId.

LeaseAccelerator also provides a field dedicated to store the ERP posting reference key “External_ Document_Id”

Also, each line in the ledger entries file has a “Status” that LeaseAccelerator will change as follows:

- Newly generated entry lines will have Status = "**New**"
- Once entries are successfully pulled by an API or pushed to an FTP folder, the status changes to "**Transferred**".
- Upon receiving the posting acknowledgments, which consists of 2 columns:
  - LeaseAccelerator posting identifier (which can be LedgerEntryLineId, PostingEntryId or PostingId)
  - The ERP posting reference key or any identifier to cross reference LeaseAccelerator with the ERP.
- The corresponding entry lines are updated so that the field "External_Document_Id" will contain the ERP posting reference key, and the Status is changed to "Posted."

Once an entry line status is "**POSTED**," LeaseAccelerator will never change or delete it, and if a change in the originating lease contract /schedule results in different ledger entries, LeaseAccelerator will generate incremental entries or delta entries as necessary but will never touch the posted entries.

LeaseAccelerator will "feel free" to delete and replace any entries that are not marked as "POSTED".

Description	DR	CR	Comments (Schedule)	LedgerEntrySubId	Status	ExternalDocumentId	PostingId
g expense	392,282.21		Schedule CyrusOne-1	28.1.1038312.1038312	NEW		LX 123
rent		392,282.21	Schedule CyrusOne-1	28.4.1038312.1038312	NEW		LX123
g expense	127,084.20		Schedule NorthPointe1	28.1.1182198.1182198	Transferred		LX555
rent		127,084.20	Schedule NorthPointe1	28.4.1182198.1182198	Transferred		LX555
rent	116,575.52		Schedule NorthPointe1	29.1.1179935.1179935	Transferred		LX555
Payable Clearing		116,575.52	Schedule NorthPointe1	29.4.1179935.1179935	Transferred		LX555
lease incenti	91.00		Schedule NorthPointe1	76.1.1187420.1187420	Transferred		LX555
entive contra expense		91.00	Schedule NorthPointe1	76.4.1187420.1187420	Transferred		LX555
g expense	11,191.70		Schedule Citizens 036	28.1.1344022.1344022	POSTED	SAP-Ref-12345678	LX090
g expense	11,191.70		Schedule Citizens 036	28.1.1344106.1344106	POSTED	SAP-Ref-12345678	LX090
g expense	11,191.70		Schedule Citizens 036	28.1.1343854.1343854	POSTED	SAP-Ref-12345678	LX090
g expense	11,191.70		Schedule Citizens 036	28.1.1343938.1343938	POSTED	SAP-Ref-12345678	LX090
rent		11,191.70	Schedule Citizens 036	28.4.1344022.1344022	POSTED	SAP-Ref-12345678	LX090
rent		11,191.70	Schedule Citizens 036	28.4.1344106.1344106	POSTED	SAP-Ref-12345678	LX090
rent		11,191.70	Schedule Citizens 036	28.4.1343854.1343854	POSTED	SAP-Ref-12345678	LX090
rent		11,191.70	Schedule Citizens 036	28.4.1343938.1343938	POSTED	SAP-Ref-12345678	LX090
rent	11,191.70		Schedule Citizens 036	29.1.1339814.1339814	POSTED	SAP-Ref-12345678	LX090
rent	11,191.70		Schedule Citizens 036	29.1.1339898.1339898	POSTED	SAP-Ref-12345678	LX090
rent	11,191.70		Schedule Citizens 036	29.1.1339982.1339982	POSTED	SAP-Ref-12345678	LX090
rent	11,191.70		Schedule Citizens 036	29.1.1340066.1340066	POSTED	SAP-Ref-12345678	LX090
Payable Clearing		11,191.70	Schedule Citizens 036	29.4.1339814.1339814	POSTED	SAP-Ref-12345678	LX090
Payable Clearing		11,191.70	Schedule Citizens 036	29.4.1339898.1339898	POSTED	SAP-Ref-12345678	LX090
Payable Clearing		11,191.70	Schedule Citizens 036	29.4.1339982.1339982	POSTED	SAP-Ref-12345678	LX090
Payable Clearing		11,191.70	Schedule Citizens 036	29.4.1340066.1340066	POSTED	SAP-Ref-12345678	LX090
g expense	€3.23		Schedule Wells Fargo-1	28.1.1018988.1018988	Transferred		

To compare using posting keys, assume your lease portfolio includes 1000 lease schedules and each lease will generate 10 lines of ledger entries. Hence the ledger entries for September 2019 will result in 10,000 lines.

LedgerEntryLineId:

- Takes the format of LLXXXXXX where XXXXXX is an integer (up to 22 characters in size)
- Unique for every line of ledger entry within the same ledger. (In the above example, you will have 10,000 lines with 10,000 different LedgerEntryLineIds).
- “Static” Once created does not change and is not affected by period closing or reopening.
- Shows in all reports once a schedule is booked regardless of the fiscal period status in LeaseAccelerator.
- The key changes with each Ledger Export generated.

The scenario for using LedgerEntryLineId:

When comparing the two posting keys, the client found out that using PostingId and splitting the file based on this key will result in 1,000 journal vouchers to be posted into the ERP GL (each will have 10 lines). The client wants to have a limited number of entries posted to their ERP GL so they decided to use the LedgerEntryLineId and include each 500 entry lines into one JV (entry to the ERP GL). And add a manual entry line to balance the journal as they are not necessarily balanced (line 501).

Pros:

- The client ended up having 20 Journal vouchers only (of 501 lines each)

Cons:

- The posting acknowledgment must be sent back for each line, so the posting acknowledgment file for LeaseAccelerator will contain 10,000 lines (each line carrying a different LedgerEntryLineId).
- Each line is either debit or credit. This means that any mistake in sending back these IDs may lead to unbalanced ledger entries in LeaseAccelerator. LeaseAccelerator has no way to control this as it is changing the status to ‘Posted’ on a line-by-line basis.
- Ledger Entries can be extracted anytime (even if the period in LeaseAccelerator is open) so nothing guarantees that users may log into LeaseAccelerator and change data while the posting roundtrip is not completed yet.
- The client had to create a fictitious entry line (line 501) in each JV to balance each batch.
- The client must apply strict manual policy / control to prevent any changes in LeaseAccelerator during the period where ledger entries are extracted until the posting acknowledgment is completed. Otherwise, the client may end up with two different versions of ledger entries; the one that was extracted and posted in LeaseAccelerator and another that was generated based on the changes that were applied to the LeaseAccelerator deals.

- The posting acknowledgment may fail if some of the lines that was extracted were deleted by LeaseAccelerator because a user applied changes to the corresponding schedule in LeaseAccelerator and the status of the entries were still not posted because the posting acknowledgment roundtrip was not completed yet. And this may cause LeaseAccelerator Posted entries to go out of balance.

PostingId:

- Takes the format of “LX”+99999 (up to 25 in size).
- Includes the “Period closing event number.”
- Is populated only when the fiscal period in LeaseAccelerator is closed.
- Every lease schedule will have at least one PostingId. If there are multiple Ledger dates or multiple currencies with the same schedules, there will be multiple PostingIds for the same schedule.
- Each PostingId will be shared among a set of balanced entries having the same ledger date and the same currency.
- The same PostingId (accordingly) marks a balanced entry.
- If the period is re-opened, the key will disappear from extracted files (except for the posted entries).
- When the period is re-closed, all lines will have new PostingIds (with the new period close event embedded). (Except for the entries already marked as posted).
- The fiscal period *closing sequence* is embedded into the PostingId key identifier. Hence every time the fiscal period is re-opened and closed, the PostingId will change. If a user tries to complete the posting acknowledgment roundtrip using key identifiers that was generated in older closing, it will be rejected.

The scenario for using PostingId

When comparing the two posting keys, the client decided to use the PostingID. Since SAP has a limitation of 999 entries per JV, the client decided to split the file to have one JV for every lease schedule (assuming the entries have the same ledger date and the same currency), resulting in 1000 JVs to be posted into the ERP GL.

Pros:

- Provides better control because:
  - This key will only be populated when the period in LeaseAccelerator is closed.
  - If the period is re-opened, the PostingIds will be wiped, meaning that if a user tries to do the posting acknowledgment for an open period it will fail.

- If the period is re-closed again, all (non-posted) ledger entries will have new PostingIds. So, if a user tries to do the roundtrip using an older version of ledger entries, it will fail as LeaseAccelerator will detect that the incoming PostingId are obsolete (refer to an older period close event).

- The posting acknowledgment file will only include 1,000 lines. Each line will have the PostingId for a complete balanced set of ledger entries representing one schedule. So even if an error occurs during posting acknowledgment (missing records for example), it will never cause LeaseAccelerator entries to go out of balance.

Cons:

- Posting to the ERP can only start when the period in LeaseAccelerator is closed.

- The number of entries to the GL ERP is higher than the ones created using LedgerEntryLineId (unless the client pack – let us say – every 10 leases in one JV)

- May constitute a risk that a large lease contract with a huge number of assets may result in very large entries that may exceed 999 lines).

PostingEntryId:

- Takes the format of "LP"+99999 (up to 25 in size).

- Works the same as PostingId above with the exception of how the PostingEntryId is assigned. In the past entries marked with the same PostingId can have thousands of rows due to adjusting entries. PostingEntryId will group rows based on the adjustment date.

- In a PostingId there are non-adjusting entries and adjusting entries. For PostingEntryId, there will only be one or the other and all rows with the same PostingEntryId will have the same adjustment date.

- This allows for smaller entries.

Integration process

Main use cases:

Common steps – with example:

1. The user closes the period in LeaseAccelerator.

2. The user generates a ledger export file from LeaseAccelerator or calls its API to generate ledger entries.

3. Integration engineers will run programs to convert them into ERP specific GL entries import document carrying the PostingId (in a proper field). Also assign a "source subledger" combined with "batch number" to all the entries (for example LA_LE_*YYYYMM*).

Use case 1: In a perfect world (Success):

1. All entries are successfully posted into the ERP GL.
2. The Integration officer runs a program to extract the “acknowledgement file” from the ERP GL that includes LeaseAccelerator PostingId and the document ID from the ERP.
3. The Integration officer calls LeaseAccelerator API or places the acknowledgement file in the FTP folder.
4. LeaseAccelerator consumes the acknowledgment file and marks the entries in its database as posted.

Use case 2: Error in posting to ERP GL -Full Batch Rejected (Best Practice):

1. Upon posting ledger entries, some ledger entries were post successfully and some fail.
2. The integration officer rollbacks ALL the posted entries, rejects the whole batch.
3. The reason for failure to be identified.
4. LeaseAccelerator period is re-opened.
5. Corrections made in LeaseAccelerator.
6. LeaseAccelerator period closed again.
7. Another batch of all the entries are re-generated and sent to the ERP GL.

Use case 3: Error in posting to ERP GL -Partial Posting:

1. Upon posting ledger entries, some ledger entries were posted successfully and some failed.
2. The integration officer runs a program to extract the “acknowledgement file” from the ERP GL that includes LeaseAccelerator PostingId and the document ID form the ERP for the posted entries only.
3. The integration officer calls LeaseAccelerator API or places the acknowledgement file in the FTP folder.
4. LeaseAccelerator consumes the acknowledgment file and marks the entries in its database as posted. (The entries that failed in posting will not be included)
5. The reason(s) for posting failure for the failed entries are investigated and identified.
6. LeaseAccelerator period is re-opened.
7. Corrections are made in LeaseAccelerator.
8. LeaseAccelerator period is closed again.

9. User generates ledger entries export again. LeaseAccelerator will exclude all entries marked as posted and will generate a complementary batch that only include the entries not marked as posted.
10. The new batch is posted to the ERP GL.
11. A new acknowledgment file is generated and sent back to LeaseAccelerator.

Note: LeaseAccelerator period was not re-opened until the posting acknowledgment round trip was completed.

Exporting ledger entries without closing the period:

Caution: The following case can cause risk of discrepancy between ERP GL and LeaseAccelerator ledger entries:

The risk occurs in any of the following cases:

1. Ledger entries (batch #1) are exported and sent to the ERP GL while LeaseAccelerator period is OPEN.
2. A user applies changes into LeaseAccelerator that affects accounting information. (Since none of the entries are marked as posted, LeaseAccelerator remove some entries and add new entries to reflect the changes).
3. Meanwhile; batch#1 entries are successfully (or even partially) posted to ERP GL.
4. The integration officer generates a posting acknowledgment file and tries to send it back to LeaseAccelerator.

Now, the following problems occurred:

1. When LeaseAccelerator tries to apply the posting acknowledgment file, it will generate errors as some of the entries posted in the ERP were deleted from LeaseAccelerator.
2. Batch #1 already fully or partially posted into the ERP GL includes wrong entries.
3. The new ledger entries export file from LeaseAccelerator cannot be posted into the ERP GL, otherwise will duplicate some entries.
4. Account balances in LeaseAccelerator and ERP GL are not properly synchronized.

To recover from this situation:

1. All batch entries posted into the ERP GL must be reversed.
2. The LeaseAccelerator support team must be called in to remove all entries for this period.
3. All ledger entries for this period in LeaseAccelerator must be regenerated.

As general rules:

- Close the LeaseAccelerator period before generating Ledger entries.
- Do not re-open the LeaseAccelerator period until the posting acknowledgment round trip is complete OR the ledger entries are totally rejected (nothing was posted to ERP GL).

Additional use cases

Case number	1
Case trigger	New entries need to be sent to the ERP general ledger for the same ledger and period as entries already sent.
Scenario	After correctly posting ledger entries to ERP general ledger, a user discovers that additional entries must be sent for the same period and for the same ledger.
Steps and Expected Outcome: A senior accounting user must: ■ Make sure all previously posted transactions are marked as POSTED in LeaseAccelerator by completing the acknowledgement roundtrip. ■ Re-open the closed period in LeaseAccelerator. ■ Perform the corrections/ updates needed. ■ The accounting user must re-close LeaseAccelerator period again. ■ For API integration, the integration officer will call the API again and process the payload. The payload shall only include the new transactions. Make sure to exclude posted entries. ■ For SFTP integration, the user should re-generate the ledger entries report manually and select the “Transfer” option, so LeaseAccelerator can deposit the file in the SFTP folder. Make sure to exclude posted entries. ■ The integration office must pick up the new file and post the new entries into the ERP GL and do the acknowledgment roundtrip.	
Warning: This option should be used carefully. If the acknowledgment roundtrip was not completed successfully (sending posting ID from the ERP to LeaseAccelerator), then some of the entries in LeaseAccelerator – though posted in the ERP – still marked as “transferred”. This means that upon doing updates and regenerating the ledger entries, these lines will be included and duplicated in the ERP.	

Case number	2
Case trigger	Client wants to automate the ledger export process.
Scenario	Client wants to configure LeaseAccelerator to automatically generate and post the ledger export file on a specific day of each month.
Steps and Expected Outcome: The ledger export report may be scheduled so that it is generated as per the stated criteria and format then delivered automatically by email on a scheduled basis, and then the transfer can be set to happen automatically on a scheduled basis. The user will need to select the criteria to only return unposted entries (ensure that posted entries are filtered out) and the correct date range and ledger. Once the ledger export file is posted in the designated folder, the general ledger integration process continues from step P400.	
Warning: Users can use the “schedule” option to set the frequency, date, and export format. When entering the report parameters, users must make sure to select the relative choices for “Starting Fiscal Year” and “Starting Period.” For example, users must select “Current Year” and “Current Month” instead of “2018” or “January”; otherwise, the system will always send January 2018 data regardless of the current month.	

Case number	3
Case trigger	GL-C2 connection is not configured, and the client wants to make sure no duplicate ledger entries can be posted to the general ledger.
Scenario	Should client decide not to send back posting acknowledgment to LeaseAccelerator, there is a risk that user mistakes may result in duplicate entries being posted to the general ledger.
Steps and Expected Outcome: Upon successful posting of ledger entries to the general ledger, The ERP archives the successfully posted entries. The client must create and run a procedure to validate ledger entries IDs (created by LeaseAccelerator) to detect if duplicates have been sent that have already been posted.	

Case number	4
Case trigger	File-based integrations – Manual transfer only. Error message is displayed in LeaseAccelerator informing user of transfer failure.
Scenario	When the user tried to transfer the ledger entries file, an error message appeared, and the transfer failed.

Steps and Expected Outcome: The user must inform the client support team. To maintain the transfer frequency and process cycle timing, the user can export the file by email or transfer to a different destination then send it to client integration officer to continue the process from step P800.
Warning: A control process must be in place to handle this situation.

Case number	5
Case trigger	API integrations only. Acknowledgment receiving issues.
Scenario	Entries were successfully posted in the general ledger in the ERP. The API used to receive the acknowledgment detects errors or warning.
Steps and Expected Outcome: Depending upon the warning policy and error policy specified in the request, LeaseAccelerator may process all or none of the matching journal entry IDs specified. Errors (invalid message, unrecognized journal entry ID) and warnings (journal entry ID previously seen) are communicated back to the ERP as part of the response payload.	

Which way to go?

LeaseAccelerator generates a set of ledger entries that should be extracted and posted to your ERP General Ledger on a monthly basis (as per the fiscal period in your ERP).

Think of LeaseAccelerator as a subledger to your lease portfolio, and just like any subledger, the fiscal period for the subledger must be closed first then the entries are extracted and posted into the ERP General ledger before the ERP GL – end of fiscal period - closing process.

The sequence should be:

- Users enter all portfolio and lease assets events while the fiscal period in LeaseAccelerator is open.
- Close the fiscal period in LeaseAccelerator to prevent any further entries / changes.
- Export ledger entries through any of LeaseAccelerator's integration methods.
- Handle the exported entries in the ERP middleware; LeaseAccelerator will export ledger entries in one file. This file can be large and include 10s of thousands of lines. So, depending on the limitations of the ERP or the company policy, clients may need to split the file into many smaller files.
- Post the files into the ERP General Ledger.
- Send the posting acknowledgment file(s) to LeaseAccelerator so that the posted entries in your ERP can be marked as "Posted" in LeaseAccelerator. This flag will prevent LeaseAccelerator

from changing or deleting these records.

- Once the acknowledgment roundtrip is completed, then users – if they need – can re- open the period to make necessary changes to the lease portfolio.
- The changes may force LeaseAccelerator accounting engine to generate different entries, but since the original entries are marked as “Posted,” LeaseAccelerator will generate incremental, additional, or corrective entries with the “delta” with status “New”.
- Users must then re-close the period
- The client integration team should now extract the non-posted entries only, so the extracted file will only include the new entries.
- Once extracted through API or FTP, the status in LeaseAccelerator will change to “Transferred.”
- The new entries should be posted to the ERP GL
- The posting acknowledgment roundtrip is repeated for these new entries.

Now let us assume your portfolio includes 1000 deals (lease contracts or schedules) and each deal has 50 assets. This portfolio generates ledger entries with 100,000 lines (debits and credits). So that when the period is closed, and the entries are extracted, the integration team will have a file of 100,000 records.

Question 1

Do you need to post the entries to multiple ERP instances?

For example, a client may have two or more ERPs - say SAP, Oracle and JDE at the same time - or may have a ledger (set of books) in LeaseAccelerator abiding to 842 standards while another ledger for European operations abiding to IFRS16.

Answer a): Yes

In the solution design you should determine how to extract the ledger entries for each instance. Maybe through having a separate ledger for each corresponding instance, or maybe having one ledger (set of books) in LeaseAccelerator while using parameters to extract the required ledger entries for each instance (such as country code, company code, business unit, any of the segments, etc.).

For file-based integration, you will need to schedule a ledger export for each instance (using the parameters in the UI) and for APIs you will need to call the APIs, once for each instance, and specify the correct request parameters to receive the correct payload.

Answer b): No

Then you would extract all ledger entries in one file. For file-based, you will schedule LeaseAccelerator to generate and export the file once and for the API method you will need to call the API once as well.

So now you have one file extracted per instance on your side.

Question 2

For each instance, will you need to split the file into smaller files (batches – or Journal vouchers)?

Answer a): No

Then you will have one Journal voucher posted in the ERP General Ledger with 100,000 debit and credit lines. If – for any reason – one line fails to post correctly, the posting process will fail, and the full batch will be rejected from the ERP.

In such case, you don't need to do the posting acknowledgment roundtrip as nothing was posted in the GL and you only need to:

- Analyze the reason for failure
- Agree on the solution
- Re-open the period in LeaseAccelerator
- Apply corrections
- Close the period
- Regenerate the ledger export
- Try again to post
- Once the posting is successful, complete the posting acknowledgment roundtrip and you should see all entries in LeaseAccelerator as Posted. (If you choose to display non-posted entries, the report will be empty).

Answer b): Yes

Then you will split the files into smaller files (Journal vouchers) and you can use the key identifier PostingId (LXnnnnn) to divide the file into smaller and balanced entries. Since PostingId groups the entries for a single deal, then – as per the example above – you will end up with 1,000 smaller files (batches).

Now the integration officers will start posting these batches into the ERP GL, one by one.

Question 2-b-1:

If any of these JVs fail to post into the ERP, will you cancel the whole posting process and roll back the previously posted JVs? (all or nothing)? Or you will allow partial posting?

As per the example above, assume one of the 1,000 Journal vouchers was addressing a retired / invalid Cost Center so the ERP rejected the Journal voucher. Will you post the 999 accepted JVs or will you reject all JVs and roll back the preceding JVs which were correctly posted in your ERP GL?

Answer 2-b-1-a) Yes, reject all the JVs and roll back the preceding ones – nothing is posted.

In such case, you don't need to do the posting acknowledgment roundtrip as nothing was posted in the GL and you only need to:

- Analyze the reason for failure
- Agree on the solution
- Re-open the period in LeaseAccelerator
- Apply corrections
- Close the period
- Regenerate the ledger export
- Re-split the files into balanced JVs
- Try again to post
- Once the posting is successful, complete the posting acknowledgment roundtrip for all the posted JVs and you should see **all** entries in LeaseAccelerator marked as Posted. (If you choose to display non-posted entries, the report will be empty).

Answer 2-b-1-b) No, we will allow partial posting

Back to the example, assume 10 JVs were rejected (failed to post in the ERP GL), and you have 980 JVs successfully posted.

In this case, you should keep the period CLOSED in LeaseAccelerator until the posting acknowledgment roundtrip is completed. Until now, LeaseAccelerator assumes that the ledger entries are not posted in the ERP since the roundtrip is not completed yet.

The users need to:

- Complete the posting acknowledgment round trip for the 980 JVs successfully posted.
- Make sure they were marked as posted in LeaseAccelerator by generating the ledger export report and EXCLUDING posted entries. The report should only include the 20 unposted JVs. (Their status should be = “Transferred”). In the same report, if the user selected to display the “Posted” entries, it should only display the 980 JVs.
- Analyze the reason(s) for failure
- Agree on the solution
- Re-open the period in LeaseAccelerator
- Apply corrections
- Close the period

- Regenerate the ledger export with parameter “Exclude Posted Entries.”
- The resulting ledger export shall only include the JVs for the unposted entries.
- Re-split the files into balanced JVs (if you need to)
- Try again to post
- Once the posting is successful, complete the posting acknowledgment roundtrip for all the posted JVs and now you should see **all** entries in LeaseAccelerator marked as Posted. (If you choose to display non-posted entries, the report will be empty).

Question 3:

If – after the period is closed in LeaseAccelerator – we need to re-open and do corrections in LeaseAccelerator?

Answer: As long as the roundtrip is completed and all ledger entries in LeaseAccelerator are marked “Posted” as per the corresponding posted entries in your ERP GL, then you can re-open the period and do any corrections. LeaseAccelerator will generate incremental entries but will not touch the posted entries.

Then you should:

- Re-close the period
- Extract the incremental entries only (by choosing to Exclude Posted Entries)
- Post the incremental entries to your ERP GL
- Compete the posting acknowledgment roundtrip to mark the new entries as posted and keep LeaseAccelerator synchronized with your ERP

Parameters for Generating Ledger Entries

Regardless of the integration method selected (RESTful APIs or file-based FTP), or as a scheduled report, LeaseAccelerator allows users to specify several parameters to generate ledger entries:

Parameter	Description	Required/Optional	Comment
As At	Report date	Required	
Starting Fiscal Year	Users can select specific year or relative values: ▪ Current fiscal year ▪ Prior fiscal year	Required	For SFTP integration, select relative value.

Parameter	Description	Required/Optional	Comment
Starting Fiscal Month	Users can specify specific month / quarter or relative month: ▪ Prior month ▪ Current month ▪ Prior quarter ▪ Current quarter	Required	For SFTP integration, select relative value.
# of Months of Lease Expense to Transfer	Number of months	Required	At least one. For integration input one.
Level of Detail	User can select details level: ▪ Portfolio level ▪ Schedule level ▪ Asset level (reporting only)	Required	For integration, only choose Portfolio or Schedule level. Asset level should only be used for reporting / BI analysis, but never for integration.
Deal Status	Allows user to include entries for deals that are incomplete or pending approval. Should never be used in integration. Default "Exclude incomplete and pending approval deals"	Do not use for integration	Includes / excludes deals incomplete or pending approval. Should only be used for reporting but never for integration.
Schedule #	If entered, only ledger entries for this schedule will be generated	Optional	Normally left black
Entity	Select Entity from a dropdown list	Optional	
Cost Center	If entered, only ledger entries for this Cost Center will be generated	Optional	Normally left black
Business Unit	If entered, only ledger entries for this Business unit will be generated	Optional	Normally left black
Country	If entered, only ledger entries for this Country will be generated	Optional	Normally left black
Lessee	If entered, only ledger entries for this Lessee will be generated	Optional	Normally left black
Set of Books	A drop down for user to a ledger.	Required	

Parameter	Description	Required/Optional	Comment
Show deals denominated in	If entered, only ledger entries for this currency will be generated	Optional	Normally left black
Exclude entries not yet transferred	Exclude entries with status = New. Default is No	Default =No	Should = No for integration
Exclude entries transferred but not yet posted	Exclude entries with status = Transferred. Default is No	Default =No	Should = No for integration
Exclude entries transferred and posted	Exclude entries with status = Posted. Default is No	Default =No	Should = Yes for integration

General ledger data mapping

As part of the general ledger integration, LeaseAccelerator both sends data to the general ledger and receives data from the general ledger.

Entries from LeaseAccelerator to general ledger

The ledger export file that LeaseAccelerator sends to the general ledger during the GL-C1 connection in the integration process includes the following fields:

Field	Type (Format)	Size	Comment
LedgerName	Text (Alphanumeric)	4000	Ledger Name
LedgerDate	Date (mm/dd/yyyy)	10	Ledger date
FXConversionDate	Date (mm/dd/yyyy)	10	Conversion date for foreign currency
FXRateType	Text		Sport or Weighted Average
TransactionalCurrency	Text	3	Transaction currency -International currency code
FunctionalCurrency	Text	3	Ledger functional currency - International currency code
ReportingCurrency	Text	3	Ledger Reporting currency - International currency code
AccountNumber	Text (Alphanumeric)	256	Natural account number in the selected ledger
AccountDescription	Text	4000	Account Name for AccountNumber (As hardcoded in LeaseAccelerator)
Segment1	Text	4000	Ledger Segment 1

Field	Type (Format)	Size	Comment
Segment2	Text	4000	Ledger Segment 2
Segment3	Text	4000	Ledger Segment 3
Segment4	Text	4000	Ledger Segment 4
Segment5	Text	4000	Ledger Segment 5
Segment6	Text	4000	Ledger Segment 6
Segment7	Text	4000	Ledger Segment 7
Segment8	Text	4000	Ledger Segment 8
Segment9	Text	4000	Ledger Segment 9
Segment10	Text	4000	Ledger Segment 10
Segment11	Text	4000	Ledger Segment 11
DrAmount	Number (decimal)	22,6	Transaction amount – Debit (see light touches)
CrAmount	Number (decimal)	22,6	Transaction amount -Credit
Comments	Text	4000	LeaseAccelerator fills this field with Schedule Number
PostingId	Text (Alphanumeric) Begins with “LX” followed by digits	16	Reference ID for LeaseAccelerator. Multiple entries will have the same PostingId. Entries sharing the same PostingId constitute a balanced entry for a schedule. Required for posting acknowledgment
PostingEntryId	Text (Alphanumeric) Begins with “LP” followed by digits		Reference ID for LeaseAccelerator. Multiple entries will have the same PostingId. Entries sharing the same PostingId constitute a balanced entry for a schedule based on adjustment date. Required for posting acknowledgment
LedgerEntryLineId	Text (LL…)	25 (Max)	Reference ID for LeaseAccelerator. Every entry (line) will have a unique LedgerEntryLineId. Required for posting acknowledgment
Status	Text		A flag for the line status: New, Transferred or Posted
JEShortDesc	Text		A short description of the accounting entry line, descriptive text.

Supported formats for file-based **exports** are: XLSX, XML, TSV, Pipe delimited and CSV.

Light-touch customizations:

LeaseAccelerator’s technical integration team offers clients a variety of modifications to the layout above to suit their specific needs.

For example, clients may request:

- Removing columns or changing the format of data.
- Adding columns with hard-coded content.
- Changing column headers.
- Removing column headers.

Additionally, for the ledger entry credit and debit amounts, light touches allow our clients to have them as any of the following forms:

- Debit amount and Credit amount in two separate columns.
- One column indicating “DR or CR” and another column for the amount.
- One Column indication “40” for Debit and “50” for Credit and another column with the amount.
- One column for the amount defaulting +ve value for Debit and -ve value for Credit (and vice versa).

Posting acknowledgement to LeaseAccelerator

When the ERP receives the entries from LeaseAccelerator for the general ledger, the ERP returns an acknowledgment to LeaseAccelerator as part of the GL-C2 connection in the integration process. The acknowledgment contains the following fields:

Field	Type (Format)	Comment
For the file-based (csv and xlsx) file the column header should be: “***Payment Reference Id***” For API, and xml file-based the field name in the xml should be: “***LedgerEntryLineID***“	Text	Reference ID that was generated by LeaseAccelerator: LedgerEntryLineId, PostingEntryId Or PostingId
ExternalDocumentId	Text	ERP ledger entries posting ID

In addition to API and SFTP, the acknowledgment file can be also uploaded through the Bulk Import feature in the user interface.

Splitting ledger entries export file:

Different ERP systems set specifications for ledger entries so that they can be ingested into its General Ledger system. Specifications probably would require some post processing to the exported ledger entries export. Ledger entries export generates one file at a time which is balanced (total debits = total credits).

Regardless of the processing and the criteria to split the file (by date, by currency, by a specific segment, etc.), each resultant group of entries must constitute balanced entry otherwise the ERP will reject it.

After splitting the Ledger Export file, integration officers may choose (or to comply with ERP specifications) to have a maximum number of lines per entry.

There are several techniques to split Ledger Entries export file into multiple files while maintaining a balanced entry. For example:

1. **Use LeaseAccelerator balanced entries (PostingID):**

Every entry in the Ledger Export file (line) will have a unique LedgerEntryLineID while a number of entries will have the same PostingID and PostingEntryID. Entries sharing the same PostingID or PostingEntryID constitute a balanced entry (Journal voucher) for one schedule.

Accordingly, upon splitting Ledger Entries file, if you maintain all the lines sharing the same PostingID or PostingEntryID together, then the resultant file will constitute a balanced entry.

This way, the Ledger export file will be split into several Journal vouchers equal to the number of deals (schedules) in LeaseAccelerator.

2. **Split balanced entries based on record count and PostingId**

After importing the Ledger Export file, integration officers may choose (or to comply with ERP specifications) to have a maximum number of lines per journal entry posted into their ERP.

For example, if the max number of lines per the new file should be 999, then integration officers can build programs to:

- Sort ledger entries by Ledger Date, Currency and PostingId or PostingEntryId (to have all entries with the same date and currency in sequence) to make sure that all entries constituting a balanced entry are grouped together.
- Count the number of entries that can be posted in one journal entry allowing a “safety” tolerance. (In the case of a max of 999 entries, the counter should import 800 lines only).
- Check the PostingId or PostingEntryId for the last imported line.
- Check the PostingId or PostingEntryId for the following line (that is not imported).
- If they are the same, then you are splitting a balanced entry by importing some lines and leaving others. Meaning the ERP would reject the Journal entry as it will not be balanced. In such case, the program should import the following lines until the PostingId or PostingEntryId changes.
- This method ensures that when the program splits the ledger entries, the imported chunks will always be balanced.

In this case, the posting acknowledgment should use the PostingId(s) or PostingEntryId(s).

3. **Split balanced entries based on record count and LedgerEntryLineId**

After importing the Ledger Export file, integration officers may choose (or to comply with ERP specifications) to have a maximum number of lines per journal entry posted into their ERP.

For example, if the max number of lines per the new file should be 999, and the integration officer decides to have 500 lines per Journal voucher only, then integration officers can build programs to:

- Sort ledger entries by Ledger Date, Currency (to have all entries with the same date and currency in sequence).
- Import 499 lines from the LeaseAccelerator- generated ledger export file (as long as the date and currency do not change).
- Add the Debit and the Credit then add line 500 as a balancing entry addressing a suspense/ settlement account in the ERP GL.
- Repeat the same process until all ledger export entries are consumed.
- Post the resulting files and the balancing entries will eventually net to zero.

Ledger Segment Integration

LeaseAccelerator requires maintaining some master data such as:

- Lessees
- Funders
- Entity
- Business Units (SBU)
- Vendors
- Cost Centers
- System users (with different roles and responsibilities)
- People included in LeaseAccelerator workflow (asset owners, asset users, etc.)
- Addresses
- GL Accounts (codes)
- Custom Participants

LeaseAccelerator allows clients to maintain master data through many options; user interface, Bulk import feature (CIW) and importing from ERPs and other sources via integration through any supported method. (See Master data integration for details).

When a client defines ledger(s), they select “Segment(s)” mapped to corresponding segments in the ERP system. LeaseAccelerator supports up to 11 segments for each ledger.

Usually, those segments are selected from available master data files, and they are called “Ledger Segments.”

Once a segment is selected for a ledger, the values for such segment can be synchronized with the sources system (ERP or other) through “Ledger Segment Integration” covered in this section.

Whenever the client retires values (entries) of the ledger segments or creates new ones, LeaseAccelerator must be synchronized to reflect the new values for the segments to maintain addressing the same values as the client’s ERP. LeaseAccelerator can be integrated with the source of this master data or segment values.

Business architecture

The client can update the ledger segments in LeaseAccelerator as needed or on a regular basis. The client must update LeaseAccelerator for each segment.

When receiving updates, LeaseAccelerator expects to receive the full set of values, not only the new ones.

For each segment received, the system will match the received values to the existing values:

- Matching values remain active; changes in name overwrite the existing name.
- New values and their corresponding names are added and activated.
- Old values not received in the update file are retired (marked inactive).

For example, a client configured LeaseAccelerator segments as follows:

Ledger Segment	Segment 1	Segment 2	Segment 3	Segment 4
Segment name	Company	Inter Company	Cost Center	Business Unit

The company may have two scenarios for updating the segment values:

- A new entry needs to be added.
- Descriptions need to be updated.

Adding a new entry

In the ERP, the client adds a new entry for the company in segment 1 (in the ERP which is – for example - mapped to the ASC 842 Ledger). The new entry has the value of 101 to represent “South Africa”. The client now needs to add this entry to LeaseAccelerator. The client must send a list of all active values for this segment, including the new entry. One of the entries should look like the following:

Source	Attribute	Name	Value
ASC 842 Ledger	company	-----	----
ASC 842 Ledger	company	South Africa	101
ASC 842 Ledger	company	-----	----

When LeaseAccelerator processes this entry, it fetches the source “ASC 842 Ledger” and the segment named “company” for a value code equal to “101”. When LeaseAccelerator does not find it, LeaseAccelerator automatically adds it.

Updating descriptions

The client decides that the official name for South Africa must be updated to Republic of South Africa and makes the update in the ERP. The same update must be made in LeaseAccelerator.

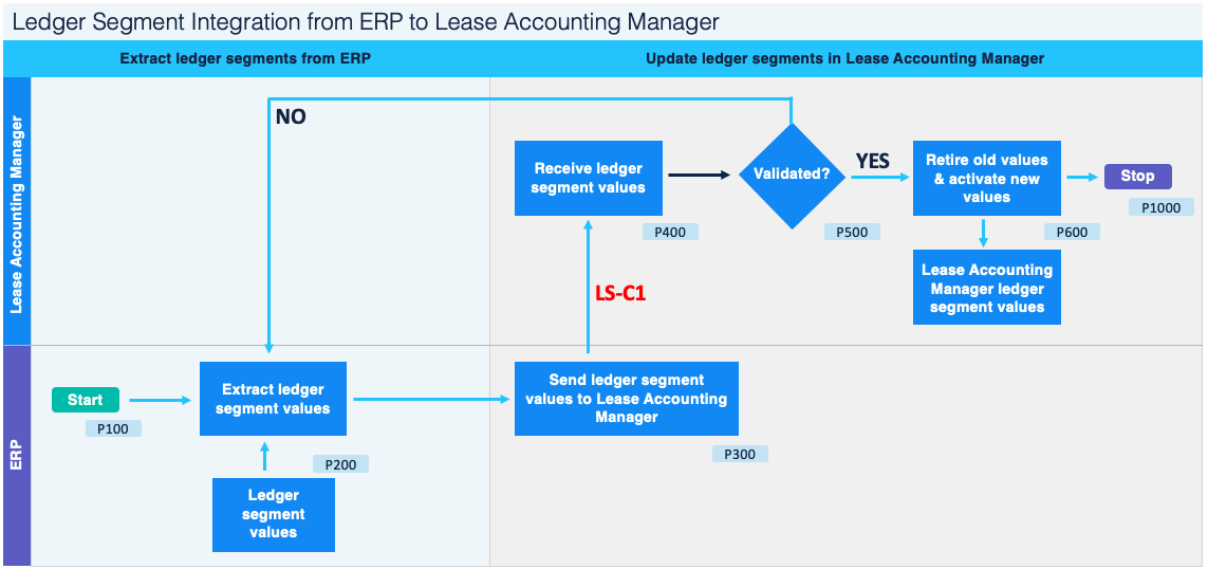
The client must send a list of **all** active values for this segment, including the updated description. One of the entries should look like the following:

Source	Attribute	Name	Value
ASC 842Ledger	company	-----	----

Source	Attribute	Name	Value
ASC 842 Ledger	company	Republic of South Africa	101
ASC 842 Ledger	company	-----	----

When LeaseAccelerator processes this entry, it fetches the source “ASC 842 Ledger” and the segment named “company” for a value code equal to “101”. LeaseAccelerator finds the record and recognizes that the name is different so LeaseAccelerator updates the name by overwriting the existing name and keeps the entry active.

Integration process



Process Step ID	Process Step Description	Phase
P100	Start	Extract Ledger Segments from ERP
P200	Extract Ledger Segments Values	Extract Ledger Segments from ERP
P300	Send LS Values to LeaseAccelerator	Update Ledger Segments in LeaseAccelerator
P400	Receive LS Values	Update Ledger Segments in LeaseAccelerator
P500	Validated?	Update Ledger Segments in LeaseAccelerator
P600	Retire Old Values & Activate New Values	Update Ledger Segments in LeaseAccelerator
P1000	Stop	Update Ledger Segments in LeaseAccelerator

P100: Start

The client's integration engine automatically triggers this process. The engine can either run the export values for each segment individually and then follow the process for each segment or the engine can run the export values for all segments at one time.

P200: Extract ledger segments values

The client's integration engine extracts the ledger segment values (entries). Each segment value can be extracted in a separate file or all segment values in one file. LeaseAccelerator can process it successfully using the Attribute field to identify the segment.

Each file must include **all** values for the exported segment(s), not just the new ones.

P300: send ledger segment values to LeaseAccelerator

Method	Process	Next Step
RESTful API	The client's integration engine calls LeaseAccelerator API (**DefineReferenceData**) with proper request data and payload. LeaseAccelerator validates and imports the payload. LeaseAccelerator acknowledges the request with an OK response.	P500
File-based	The client's integration engine places the file into the client's SFTP folder then to the client's DMZ area then to LeaseAccelerator's SFTP folder.	P400

P400: Receive ledger segment values files

LeaseAccelerator's file-watcher automatically detects the Ledger Segment file. (Note that Ledger Segment values can also be uploaded manually using the Bulk Import functionality in the LeaseAccelerator application.)

P500: Validated?

LeaseAccelerator inspects the file for errors and potentially invalid entries.

Method	Process	Next Step
RESTful API	If the file contains any errors, LeaseAccelerator rejects it and sends error status in the API Response. The ledger segment value must be re- extracted, and the API called again. (P200) Otherwise, LeaseAccelerator acknowledges the request with an OK response.	P600
File-based	If the file contains any errors, LeaseAccelerator rejects it and sends an email alert to the designated person. (P200).	P600

P600: Retire old values & activate new values

LeaseAccelerator matches the received entries to the existing ones as follows:

- Matching values remain active, changes in name overwrite the existing name.
- New values and their corresponding names are added and activated.
- Old values not in the Ledger Segment file are retired and marked inactive.

LeaseAccelerator sends an email to the designated person.

P1000: Stop

The process ends, and the client updates the proper log.

Additional use cases

Cast trigger	The client needs to send one file, instead of five files.
Scenario	The client needs to send one file to update all segments, instead of sending one file for each ledger segment.
Steps and Expected Outcome: The client combines the five files into one file and LeaseAccelerator processes it successfully using the Attribute field to identify the segment.	

Integration data

During the integration process, LeaseAccelerator receives data from the Ledger Segment file as shown in the LS-C1 connection in the integration process diagram on page 49.

Ledger segments to LeaseAccelerator

LeaseAccelerator expects the following data structure:

Field	Type (Format)	Allowed to be Null	Comment
Source	Text	no	The source ledger from ERP. Sending empty ledger name updates all ledgers.
Attribute	Text	no	The segment name (company, cost center, etc.)
Name	Text	no	The segment entry name (USA, South Africa, etc.)
Value	Text	no	The segment entry value (usually digits, 101, 144, etc.)

If you don’t specify the source ledger to use to resolve the attribute you are updating in the received file, the LeaseAccelerator importer chooses the oldest unretired ledger which is “Default Ledger” and searches for a segment name that matches the sent “attribute.”

Accounts Payable Integration

Business architecture

Companies use the accounts payable system in the ERP to determine how and when to properly allocate cash disbursements and to process payments. LeaseAccelerator does not duplicate your ERP AP system. It maintains all the necessary information regarding lease agreements, assets, payment terms and schedules, accounting standards and information about client sets of books and ledger segments to calculate the timing and amounts of due payments to lessors (funders).

Synchronizing LeaseAccelerator with the accounts payable system ensures that the accounts payable team is armed with the necessary details required to approve, dispute, and disburse payments to lessors. LeaseAccelerator is not a billing / invoicing system. It is not intended to generate or maintain invoices, credit notes, payment instructions, discounts, payment terms, supplier bank accounts, etc.

However, clients may elect to use the due payments to generate invoices and post them into their ERP accounts payable module and to process the payments accordingly. Clients must set the necessary procedures and controls to review, examine and verify the generated due payments before and after converting them into invoices and should apply the client's standard invoice approval / payment policies and procedures. In such a case, the client does not expect to receive invoices from lessors.

LeaseAccelerator AP integration also includes the roundtrip of importing actual disbursements (paid through the ERP AP module) and matching them to the corresponding due payments.

This step is important to balance the AP clearing account in LeaseAccelerator.

Due Payments

LeaseAccelerator can generate due payments:

- On a monthly or weekly basis
- For a specific ledger (set of books)
- In schedule transactional currency (not converted to ledger currency if different)
- At the lease (schedule or deal) level or at portfolio level
- Optionally filtered by selecting any combination of available parameters (segments, tags, currency, schedule, lessee, country, entity, asset type, business unit)
- Optionally excluding evergreen assets
- Optionally excluding payments that have not been transferred before to the ERP
- Optionally excluding entries transferred and paid

LeaseAccelerator has its own accounting engine. Accordingly, it automatically generates all expense and balance sheet recognition entries. These entries are posted to LeaseAccelerator's subledger clearing account.

Actual disbursements:

Actual disbursements are processed through the ERP Accounts Payable module and posted to the accounts payable clearing account on the ERP general ledger.

LeaseAccelerator receives updates from the ERP's accounts payable to capture the actual disbursements made for each lease. The updates can be sent at any frequency.

Actual disbursements may vary in amount and date paid from due payments calculated by LeaseAccelerator for a variety of reasons. Variances can result from either:

- **One-time / variable change**, such as late payment fee, over-usage of equipment, etc. which will not affect successive payments.

- **Permanent change**, which could result in any change in terms due to renegotiated contracts or changes in leased assets status (buyout, terminated, etc.) or any other conditions remain effective for future payments. Clients must resolve these discrepancies by updating LeaseAccelerator terms to keep LeaseAccelerator synchronized and aware of correct payments.

- **Late payments or prepayments** where the client skipped a payment in a previous month so the actual disbursement for that month was zero and the following month it is doubled. Or simply if – for any reason – the client decides to pay in advance.

The resulting net balance in the LeaseAccelerator subledger clearing account represents a timing or amount difference between the due payment and the actual disbursement. Clients have two options for handling this net balance: roll this account into the desired line item for financial statement presentation or record a general ledger entry to transfer the net balance in the clearing account to either prepaid or actual as needed, which would not be performed in the subledger.

Upon receiving the actual disbursements, LeaseAccelerator can match each of them to the corresponding due payment either by:

- **LedgerEntryLineId:** LeaseAccelerator generates a unique reference key identifier for every due payment line. This key is called "LedgerEntryLineId" which cannot be repeated across the lifetime of the client database. Clients who decide to import LeaseAccelerator's due payments and turn them into AP invoices in their ERP should keep track of this key and send it back along with the corresponding actual disbursement(s) so that LeaseAccelerator can match the actual disbursements to the corresponding due payments. Credit notes can be sent back to LeaseAccelerator with negative amounts.

- **Other key identifiers:** Clients who decide to process lease payments based on lessors' invoices or other sources of payments or recurring payment generator will not be able to import the unique key identifier generated by LeaseAccelerator. In such case upon receiving the actual disbursements, LeaseAccelerator will try to match the actual disbursements to their due payments based on a combination of keys:
  - Currency
  - Due date (month and year)
  - Purchase Order Number

- If duplicate records were found, LeaseAccelerator will try to match based on Lessor Reference Id

If you are planning to use this method, make sure PO numbers and Lessor Reference Id's are correctly populated in both LeaseAccelerator and your ERP system.

The specific integration model for the LeaseAccelerator accounts payable is determined by which business requirements are applicable from the following considerations:

- LeaseAccelerator due payments only
- Lessor invoices only
- Lessor invoices as payments

Due Payments Only Model

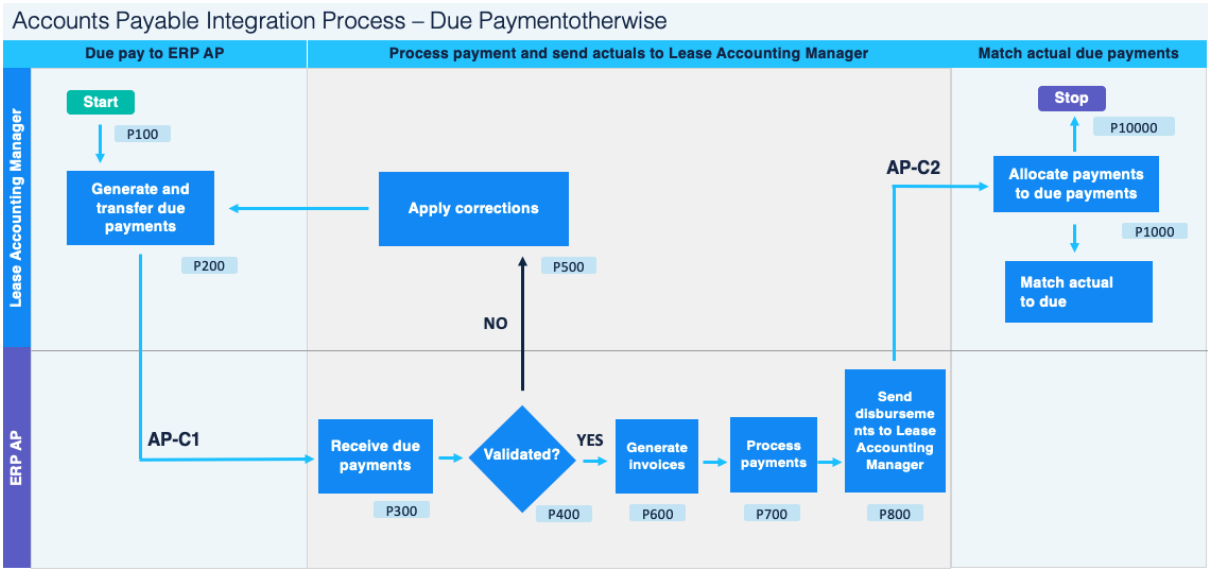
In this model, LeaseAccelerator generates due payments information based on the contractual terms of the lease. LeaseAccelerator is not a billing / invoicing system. It is not intended to generate or maintain invoices, credit notes, payment instructions, discounts, payment terms, supplier bank accounts, etc.

However, clients may elect to use the due payments to generate invoices and post them into their ERP accounts payable module and to process the payments accordingly. The client does not expect to receive invoices from lessors.

Actual payments need to be transferred back to LeaseAccelerator to adjust the clearing account in LeaseAccelerator.

Disclaimer:

Clients who elect to use LeaseAccelerator generated due payments to generate invoices will do this process at their own discretion and under their full responsibility without any liability toward LeaseAccelerator. Clients must set the necessary procedures and controls to review, examine and verify the generated due payments before and after converting them into invoices and should apply the client's standard invoice approval / payment policies and procedures.



Process #	Process Step Description	Phase
P100	Start	LeaseAccelerator Due Pay to AP in ERP
P200	Generate & Transfer Due Payments	LeaseAccelerator Due Pay to AP in ERP
P300	Send Due Payment to AP in ERP	LeaseAccelerator Due Pay to AP in ERP
P400	Valid?	Process payment & send Actuals to LeaseAccelerator
P500	Apply Corrections	Process payment & send Actuals to LeaseAccelerator
P600	Generate Invoices	Process payment & send Actuals to LeaseAccelerator
P700	Process Payment	Process payment & send Actuals to LeaseAccelerator
P800	Match Payments to Due Payments	Reconcile payment with Due Payments
P800	Send Disbursement to LeaseAccelerator	Process payment & send Actuals to LeaseAccelerator
P1000	Allocate Payments to Due Payments	Reconcile payment with Due Payments
P10000	Stop	Reconcile payment with Due Payments

P100: Start

As part of monthly or weekly accounting activity, due payments to lessors (funders) need to be transferred from the LeaseAccelerator subledger to accounts payable in the ERP as invoices.

P200: Generate and send due payments

Depending on the integration method selected, AP-C1 should be activated on a weekly or monthly basis to generate and transfer due payments:

Method	Process	Next Step
RESTful API	ERP integration adapter initiates an API call to LeaseAccelerator to generate a Payables Export report. LeaseAccelerator generates an XML response containing the due payments for the requested week or month.	P300
File-based	Based on the configured schedule for AP due payments, LeaseAccelerator shall automatically generate and send Due payments and transfer the file in the desired format into the SFTP folder. Users can also manually generate Due payments through the Reporting Integration UI and select to transfer the export file.	P300

Each entry has a unique payment identifier (LedgerEntryLineId) so that LeaseAccelerator can track the entry and match actual payments to due payments based on that identifier.

LeaseAccelerator stores the transmitted due payments with LedgerEntryLineId, time/date of request, request parameters, and user requesting the transfer.

P300: Receive due payments (AP-C1)

Method	Process	Next Step
RESTful API	ERP integration adapter receives the API response payload (due payments)	P400
File-based	In the agreed upon file exchange destination folder on the client- provided SFTP server, a mechanism detects the new file and initiates ingest of the file.	P400

P400: Valid?

The accounts payable user reviews and approves the due payments imported from LeaseAccelerator. The user must account for any of the following cases:

- One-time additional fees or deductions due to events pertaining to this particular period
- A re-negotiated term affecting the payments for this and recurring periods
- An event that was not recorded in LeaseAccelerator (on asset, schedule, or portfolio levels)

If due payments are valid proceed to step P600, otherwise proceed to step P500.

P500: Apply corrections

The accounts payable accountant informs the assigned LeaseAccelerator user who corrects the data entered in LeaseAccelerator and regenerates the Due Payments as per the method selected:

Method	Process	Next Step
RESTful API	ERP integration adapter initiates an API call to LeaseAccelerator to generate a payable export message.	P300

Method	Process	Next Step
File-based	Users manually generates Due payments through the Reporting Integration UI and select to transfer the export file.	P300

P600: Generate invoices

The integration officer runs a program to convert due payments into invoices and posts the invoices into the ERP AP module. Generated invoices should be processed as per normal payable approval and payment process.

Due payments key identifier (LedgerEntryLineId) must be maintained in the invoices.

Generated invoices must carry a flag that identifies them as lease invoices.

P700: Process payments

The accounts payable officers process the payment according to the client's policies and procedures.

Payments must carry the key identifier (LedgerEntryLineId). A single invoice may be paid by one or more payments.

P800: Send disbursements to LeaseAccelerator (AP-C2)

As a trigger or as a nightly process, the ERP recognizes posting of lease payments to accounts payable.

The ERP generates a file with actual disbursements (corresponding to lease invoices only) containing the information that must be sent to LeaseAccelerator including the key identifier (LedgerEntryLineId).

Method	Process	Next Step
RESTful API	ERP integration adapter initiates an API call to LeaseAccelerator with the generated payload (actual disbursements). LeaseAccelerator validates and imports the payload.	P1000
	LeaseAccelerator acknowledges the request with an OK response.	
File-based	The integration officer places the actual disbursements file in the agreed upon FTP folder. LeaseAccelerator's ingest task fires on a configured periodicity (as often as every ten minutes; as infrequently as weekly) and checks the configured integration point for new files. Recognizing that a new, unprocessed file has arrived, LeaseAccelerator validates and imports the file. LeaseAccelerator sends an email notification to the configured recipients that the file has been successfully received and processed. LeaseAccelerator removes the detected file from the SFTP location.	P1000
Bulk Import	Users access Bulk Import feature in LeaseAccelerator and manually imports the disbursements file.	P1000

P1000: Match Actual payments to due payments

LeaseAccelerator applies the actual payments to the unpaid due payments according to the keys supplied (LedgerEntryLineId) within the actual disbursements file. Refer to the “LeaseAccelerator Application of Actual Payments to Due Payments” section below for a detailed explanation.

Additional use cases

Case trigger	Lessor disputes payment amounts
Scenario	The client receives a contact from a lessor disputing the processed payment.
Steps and Expected Outcome: A lease accountant reviews lessor justification for disputing the paid amounts. A discrepancy can result from: ▪ One-time charge, such as late payment fee, over-usage of equipment, etc. which will not affect successive payments. In this case, a LeaseAccelerator user must use payment adjustment or related expense features in LeaseAccelerator and re-send the due payments (selecting the new payments only). ▪ Change in payment terms and conditions, which could result in any change in terms. In this case, a LeaseAccelerator user must correct the corresponding lease terms and conditions and re-send the due payments (selecting the new payments only).	
Warning: If Month-End Close is in use, and the payment is adjusted after closing the books for the month, two entries may be sent: a reversing entry to “undo” the previously posted entry to the accounts payable clearing, and a correction entry to post the revised payment amount to the accounts payable clearing. The net of these will capture the difference in payment amount.	

Lessor Invoice-Only Model

This model is usually used for any of the following cases:

- In a transitional period of implementing LeaseAccelerator when not all data entered in LeaseAccelerator are verified.

- For equipment where payments are dependent on actual usage or other variable factors. In such a case, clients enter the base lease data in LeaseAccelerator and process the payments based on lessors' invoices.

The client decides to proceed and pay lessors' invoices. However, actual payments should be sent back to LeaseAccelerator.

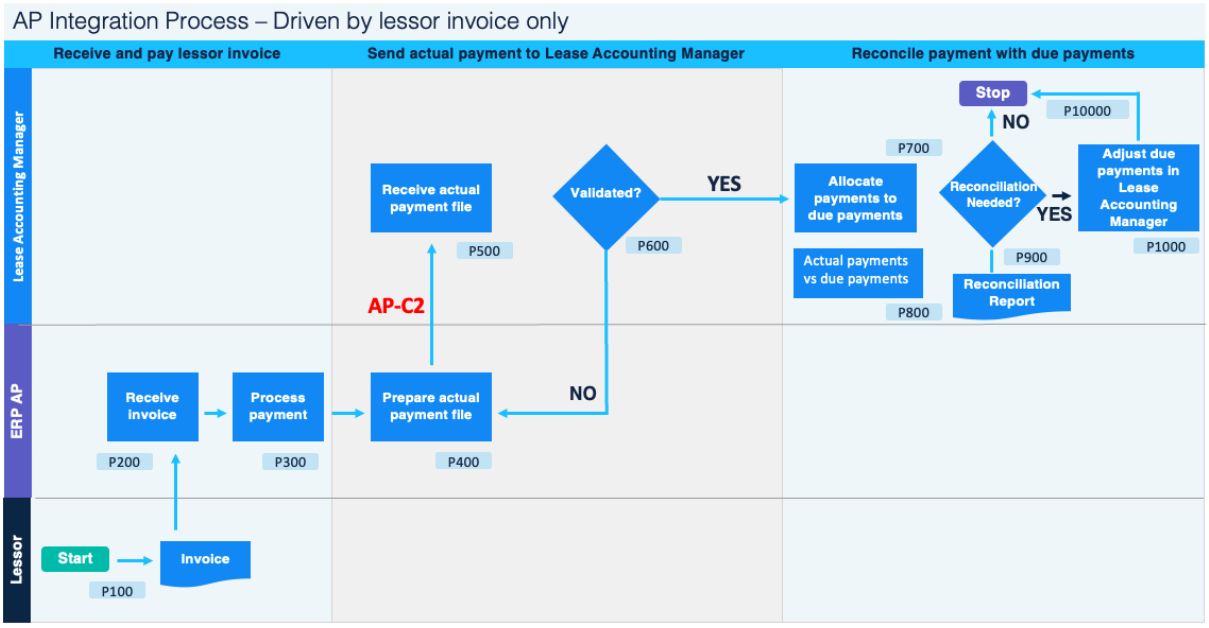
This integration model has the following consequences:

- **Duplicate accounting effort and possibility of discrepancy:** LeaseAccelerator has all the lease accounting related information, EoT terms, accounting standards, company ledgers and segments, cost center, asset users and owners, etc. LeaseAccelerator generates the correct amounts allocated to the proper segments according to the correct accounting classification rules. Lessor invoices do not carry this information. The accounts payable accountant must determine the needed accounting entries.

- **No pre-payment control:** Since due payments are not exported from LeaseAccelerator to accounts payable in the ERP, accounts payable accountants have no automated means to match or verify lessors' invoices. Any discrepancy between paid invoices and LeaseAccelerator calculated due payments will appear when actual payments are sent back to LeaseAccelerator. In this case, the LeaseAccelerator clearing account will show "after the fact" residual balances.

- **LeaseAccelerator clearing account reconciliation limits:** Since due payments are not exported from LeaseAccelerator to accounts payable in the ERP, the payment key identifier that LeaseAccelerator generates for each entry is not to be delivered to accounts payable in the ERP. When accounts payable sends back the actual payments to LeaseAccelerator, they should include other keys to enable LeaseAccelerator to correctly match and allocate these actual payments to the due payments. This limits client flexibility in tracking intentional changes in payments, for example, being able to intentionally skip a payment and leave that specific payment period marked as unpaid until a dispute is resolved.

- **Lease payment identification risk:** The key identifier sent from LeaseAccelerator to accounts payable for each due payment entry helps the accounts payable user to filter and send information for only lease payments to LeaseAccelerator. When sending back the actual payments to LeaseAccelerator, the transfer will then only include lease-related payments. When the key identifier is not used, as in this scenario, the accounts payable users must be very cautious to manually select only the lease-related payments.



Process Step ID	Process Step Description	Phase
P100	Start	Receive and pay lessor invoice
P200	Receive Invoice	Receive and pay lessor invoice
P300	Process Payment	Receive and pay lessor invoice
P400	Prepare Actual Payment File	Send Actual Payment to LeaseAccelerator
P500	Receive Actual Payments File	Send Actual Payment to LeaseAccelerator
P600	Validated?	Send Actual Payment to LeaseAccelerator
P700	Allocate Payments to Due Payments	Reconcile Payment with Due Payments
P800	Actual Payments vs. Due Payments	Reconcile Payment with Due Payments
P900	Reconciliation Needed?	Reconcile Payment with Due Payments
P1000	Adjust Due Payments in LeaseAccelerator	Reconcile Payment with Due Payments
P10000	Stop	Reconcile Payment with Due Payments

P100: Start

This process must be triggered regularly to apply actual payments to due payments and control the LeaseAccelerator clearing account.

P200: Receive invoice

Lessors send their invoices regularly based on the leasing contractual terms and conditions. Invoices may include additions or deductions reflecting specific situations or conditions pertaining to the invoiced period.

P300: Process payment

Accounts payable officers review lessors' invoices and process the payment accordingly. Users are encouraged to match the invoices against LeaseAccelerator due payments to proactively prevent erroneous payments to lessors and to correct wrong information sent to LeaseAccelerator.

P400: Prepare Actual Payment File

On a regular basis, the account payable integration officer extracts the actual payments. In this case, users must be cautious to include lease-related payments only.

P500: Receive actual payments file

Method	Process	Next Step
RESTful API	ERP integration adapter initiates an API call to LeaseAccelerator with the generated payload (actual disbursement). LeaseAccelerator validates and imports the payload. LeaseAccelerator acknowledges the request with an OK response.	P600
File-based	The integration officer places the actual disbursement file in the agreed upon folder. LeaseAccelerator's ingest task fires on a configured periodicity (as often as every ten minutes; as infrequently as weekly) and checks the configured integration point for new files. Recognizing that a new, unprocessed file has arrived, LeaseAccelerator validates and imports the file. LeaseAccelerator sends an email notification to the configured recipients that the file has been successfully received and processed. LeaseAccelerator removes the detected file from the SFTP location.	P600

P600: Validated?

LeaseAccelerator validates the file for format and content. In the event of any error, LeaseAccelerator reverts any updates processed for the file and sends an email notification to the configured recipients advising of the failure.

P700: Allocate payments to due payments

LeaseAccelerator applies the actual payments to the unpaid due payments (calculated by LeaseAccelerator) according to the keys supplied within the actual disbursement file. Refer to the "LeaseAccelerator Application of Actual Payments to Due Payments" section below for a detailed explanation.

P900: Reconciliation needed?

A LeaseAccelerator user generates the accounts payable reconciliation report and detects any discrepancies between LeaseAccelerator calculated due payments and actual payments imported from the ERP. The accounts payable clearing account balance should only reflect unpaid due payments and pre-paid actual payments. If reconciliation is needed, proceed to step P1000. If no reconciliation is needed, the process ends.

P1000: Adjust due payments in LeaseAccelerator

If the actual payments do not match the LeaseAccelerator calculated due payments, the user must investigate the discrepancy and use the proper features in LeaseAccelerator to match the payments. This step should be repeated until full reconciliation is achieved.

Additional use cases

Case trigger	Duplicate schedule numbers for different lessors.
Scenario	Since the client did not export the due payments from LeaseAccelerator to accounts payable in the ERP, accounts payable does not have the payment ID. In sending the actual payments to LeaseAccelerator, the client relies on other keys such as schedule number, purchase order number, or lessor (payee) code. If the schedule numbers or purchase order numbers are duplicated for different lessors, LeaseAccelerator may allocate the actual payments to wrong lessors.
Steps and Expected Outcome: The client must make sure that the combination of schedule number and purchase order number cannot be duplicated for different lessors.	

Case trigger	Actual payments exceed due payments.
Scenario	Actual payments sent from the ERP exceed the total due payments calculated by LeaseAccelerator.
Steps and Expected Outcome: LeaseAccelerator allocates the actual payments to due payments based on the following sequence (when applicable): ▪ LeaseAccelerator allocates the excess amounts as per the due date specified. ▪ LeaseAccelerator allocates excess payments to any unpaid due payments in previous periods. ▪ LeaseAccelerator allocates excess payments to future due payments. This sequence properly accounts for late payments and prepayments.	

Case trigger	Actual Payments Due date missing or does not match any payment due date.
Scenario	Actual Payments sent from the ERP do not have a valid payment due date.
Steps and Expected Outcome: LeaseAccelerator allocates the actual payments to due payments to any unpaid due payments in previous periods. LeaseAccelerator allocates excess payments to future due payments. This sequence properly accounts for late payments and prepayments.	

Case trigger	Credit note received from lessor.

Scenario	The client relies only on lessors' invoice for payment. After payment has been processed and actual payment sent back to LeaseAccelerator, the client discovered that the invoice was incorrect and disputed it with the lessor.
Steps and Expected Outcome: When an invoice is proved to be incorrect, the lessor sends a credit note to cancel the effect of the disputed invoice then sends a new invoice with the correct amount. After Month-End Close, record a Payment Adjustment in LeaseAccelerator to capture the corrected invoice amount. A reversing entry will be created to reverse the previous posting to the accounts payable clearing account, and a corrected entry will be added to capture the revised invoice amount. The client then needs to use the payables export report to generate the data required to populate an Import Disbursements file which can be manually completed to capture the refund (if issued by the lessor) or credit note and adjust the accounts payable clearing account balance accordingly.	
Warning: LeaseAccelerator strongly recommends that lessor invoices are matched against due payment report in LeaseAccelerator before approving payments.	

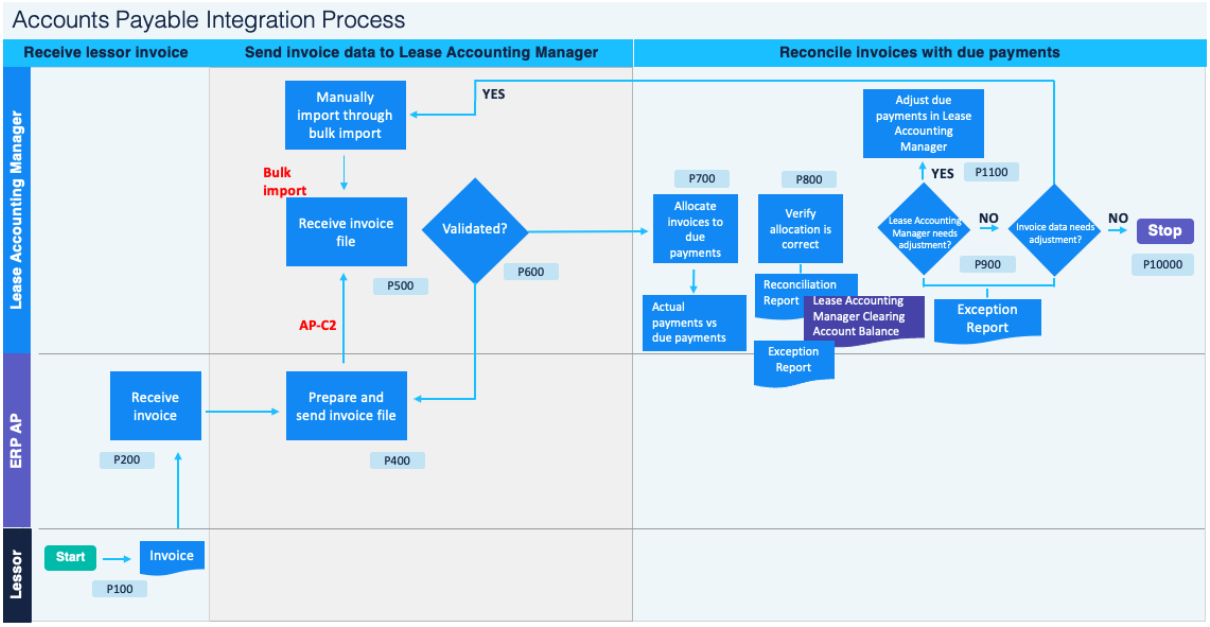
Lessor Invoices as Payments

In this model, clients sent accounted accounts payable invoices to LeaseAccelerator as if they were actual payments. LeaseAccelerator matches the accounts payable invoices with due payments for reconciliation of the accounts payable clearing account as follows:

- LeaseAccelerator receives the invoice feed from accounts payable as deemed payment. From the LeaseAccelerator perspective, this feed is the standard payment feed.
- Once “payment” is matched with accounts payable clearing account in LeaseAccelerator then LeaseAccelerator generates a non-posting entry: Dr AP Clearing, Cr Cash.
  - The above entry/matching can be leveraged for LeaseAccelerator accounts payable reconciliation to determine what’s “paid” and what’s not.
  - It also depletes the accounts payable clearing account balance in the BI accounting reports.
  - However, it is not reflected in the ledger export report, and is not posted to the general ledger.
- The standard entry crediting the accounts payable clearing account is posted to the general ledger, where it will be offset against the debit to the accounts payable clearing account received from the accounts payable subledger.
- The accounts payable feed, and “payment” matching occur in transaction currency. LeaseAccelerator does not perform any foreign exchange (FX) remeasurement or translation as part of this process.
- The non-posting entry has no impact on the disclosure reports, which use assume paid approach.
- If BI accounting reports are used as part of the accounts payable reconciliation, they should always be run in transaction currency. The results in functional currency or reporting currency are not reliable for this purpose.

LeaseAccelerator sends an exception report with unmatched payments. The client’s business team receives the exceptions and must have the ability to fix the exceptions and load the corrected data into LeaseAccelerator for matching.

Integration Process



Process Step ID	Process Step Description	Phase
P100	Start	Receive lessor Invoice
P200	Receive Invoice	Receive lessor Invoice
P400	Prepare and Send Invoice file	Send Invoice data to LeaseAccelerator
P500	Receive Invoice file	Send Invoice data to LeaseAccelerator
P600	Validated?	Send Invoice data to LeaseAccelerator
P700	Allocate Invoices to Due Payments	Reconcile Invoices with Due Payments
P800	Actual Payments vs. Due Payments	Reconcile Invoices with Due Payments
P800	Verify Allocation Is Correct	Reconcile Invoices with Due Payments
P900	LeaseAccelerator Needs Adjustment?	Reconcile Invoices with Due Payments
P1000	Invoice Data Needs Adjustment?	Reconcile Invoices with Due Payments
P1100	Adjust Due Payments in LeaseAccelerator	Reconcile Invoices with Due Payments
P1200	Manually Import through Bulk Import	Send Invoice data to LeaseAccelerator
P10000	Stop	Reconcile Invoices with Due Payments

P100: Start

This process can be run as frequently as the client decides.

P200: Receive invoices

Lessors send their invoices regularly based on the leasing contractual terms and conditions. Invoices may include additions or deductions reflecting specific situations or conditions pertaining to the invoiced period.

P400: Prepare & send invoice file

Based on a scheduled configuration, the client's integration engine generates a file with invoice data as per the agreed format.

Method	Process	Next Step
RESTful API	ERP integration adapter initiates an API call to LeaseAccelerator with the generated payload (invoices as actual disbursements). LeaseAccelerator validates and imports the payload. LeaseAccelerator acknowledges the request with an OK response.	P600
File-based	The client's integration engine places it in the DMZ zone.	P500

P500: Receive invoice file

LeaseAccelerator's ingest task fires on a configured periodicity (as often as every ten minutes; as infrequently as weekly) and checks the configured integration point for new files.

Recognizing that a new, unprocessed file has arrived, LeaseAccelerator validates and imports the file.

LeaseAccelerator sends an email notification to the configured recipients that the file has been successfully received and processed.

LeaseAccelerator removes the detected file from the SFTP location.

Invoice data can also be uploaded into LeaseAccelerator via the Actual Disbursement Bulk Import function in the LeaseAccelerator application.

P600: Validated?

LeaseAccelerator validates the format and content for major errors. If the file is corrupt, the whole file is rejected. LeaseAccelerator expects to find key identifiers such as PO number and due date. LeaseAccelerator records entries missing key identifiers. Any record that does not have all key identifiers is rejected.

If the file being validated was uploaded through the Bulk Import functionality in the LeaseAccelerator application, the LeaseAccelerator application displays the records and the status of each one. The user may elect to proceed and upload valid records or abort.

If LeaseAccelerator automatically uploaded the file being validated, LeaseAccelerator sends an email to the designated user indicating the upload status for each record.

If the file or some records are rejected, go to step P400. Otherwise proceed to step P700.

P700: Allocate invoices to due payments

LeaseAccelerator automatically allocates the invoices to the unpaid due payments (calculated by LeaseAccelerator) according to the keys supplied within the actual disbursement file. Refer to the "LeaseAccelerator application of Actual Payments to Due Payments" section for a detailed explanation.

If LeaseAccelerator fails to allocate any records of actual payments to proper due payments (such as the keys or amounts do not match), it generates an exception report that contains those records. The report, which is in Microsoft Excel format, is emailed to the designated user. LeaseAccelerator does not import any records in the exception report into its database.

The user must correct the errors and reload the corrected records through the Bulk Import functionality in the LeaseAccelerator application as per step P1200.

Proceed to step P800.

P800: Verify allocation is correct

The LeaseAccelerator user generates accounts payable reconciliation reports and a LeaseAccelerator accounts payable clearing account report for use in investigating the records in the exception report. The user determines the reason LeaseAccelerator failed to allocate each of the records to due payments.

P900: LeaseAccelerator needs adjustment?

If – for any invoice entry – LeaseAccelerator due payments need to be adjusted due to inaccurate data fed into it, proceed to step P1100. Otherwise proceed to step P1000.

P1000: Invoice data needs adjustment?

If LeaseAccelerator failed to allocate an invoice to a due payment because of invalid keys or wrong invoice amount, proceed to step P1200. Otherwise, reconciliation is complete, and the process stops.

P1100: Adjust due payments in LeaseAccelerator

If the actual payments do not match the LeaseAccelerator calculated due payments, the LeaseAccelerator user must investigate any discrepancies and use the features in the LeaseAccelerator application to match the payments.

This step must be repeated until full reconciliation is achieved.

P1200: Manually import through bulk import

The client prepares the corrected exception report and manually uploads it into LeaseAccelerator through the Actual Disbursements Bulk Import function in the LeaseAccelerator application.

P10000: Stop

The integration ends, and the client updates the corresponding log.

Additional use Cases

Case trigger	Amount of invoice received does not match the corresponding due payment.
Scenario	LeaseAccelerator matches an invoice to a due payment but the amount of the invoice either exceeds or is less than the corresponding due payment.
Steps and Expected Outcome: LeaseAccelerator rejects the invoice, directing it to the exception report, and informs the assigned team. The assigned team contacts the accounts payable team and/or lease contracts management team for vendor follow up. Once the mismatch reason is resolved, a LeaseAccelerator user manually uploads the invoice using the Actual Disbursement Bulk Import functionality in the LeaseAccelerator application.	

Case trigger	Invoice received by accounts payable earlier than due payment.
Scenario	The client received an invoice from a lessor before LeaseAccelerator generated the due payment for this fiscal period.
Steps and Expected Outcome: LeaseAccelerator fails to match the invoice as the payment is not due yet. For example, a lessor sends an invoice in January requesting a payment due in February.	
Warning: If schedule terms stated pre-payments (payments for February will be billed in January) then the client must configure the payment date in LeaseAccelerator to recognize this schedule, so due payment for February appears in January and LeaseAccelerator is able to successfully match it.	

Case trigger	Lease invoice receipt sent later with consolidated payment.
Scenario	The client did not receive an invoice due for a specific period. In the following period, the lessor sent a consolidated invoice covering the current and the previous (unbilled) period.
Steps and Expected Outcome: LeaseAccelerator rejects the invoice since the amount exceeds the due payment. Using the exception report, the client must split the invoice into two invoices with the correct due payments and corresponding amounts then upload them manually into LeaseAccelerator.	

Case trigger	Lease schedule payment matches.
Scenario	Lease payment for real estate with multiple schedules where total lease payment schedule amount equals total accounts payable invoice amount for the same month.
Steps and Expected Outcome: The client sends PO number and not schedules, so matching in LeaseAccelerator must be based on PO number and due date. If the total of the payments for all LeaseAccelerator schedules for the specified PO matches the specified payment and the due dates, LeaseAccelerator accepts and allocates the payment to the schedules.	

Case trigger	Lease schedule payment more.
Scenario	Lease payment for real estate with multiple schedules where the total lease payment schedule amount is greater than the total accounts payable invoice amount for the same month.
Steps and Expected Outcome: The client sends PO number and not schedules, so matching in LeaseAccelerator must be based on PO number and due date. If the total of the payments for all LeaseAccelerator schedules for the specified PO does not match the specified payment and the due dates, LeaseAccelerator rejects the specified payment.	

Case trigger	Accounts payable invoice has not match due to invalid PO number.
Scenario	LeaseAccelerator fails to match the invoice because the PO does not exist in LeaseAccelerator.
Steps and Expected Outcome: LeaseAccelerator rejects the invoice and adds it to the exception file. If the PO was incorrectly input in LeaseAccelerator, then it must be corrected, and the invoice manually reloaded. If this PO is for a non-lease business transaction, then this record must be deleted from the report and ignored.	

Case trigger	Credit note received from lessor.
Scenario	The client receives credit note from lessor after payment has been processed and actual payments were sent to LeaseAccelerator.
Steps and Expected Outcome: An invoice should be sent to LeaseAccelerator with negative numbers.	

Integration Data

Due payment from LeaseAccelerator to ERP AP: AP-C1

LeaseAccelerator will export the following information when the due payment report is transferred or as a response payload when the API is invoked.

Field	Type	Width	Comment
LedgerDate	Date	10	Payment Due date
AccountNumber	Text	256	Natural Account Number
AccountDescription	Text	4000	Natural Account Description
LedgerName	Text	4000	Ledger Name
Segment1	Text	4000	As per the Ledger configuration
Segment2	Text	4000	As per the Ledger configuration
Segment3	Text	4000	As per the Ledger configuration
Segment4	Text	4000	As per the Ledger configuration
Segment5	Text	4000	As per the Ledger configuration
Segment6	Text	4000	As per the Ledger configuration
Segment7	Text	4000	As per the Ledger configuration
Segment8	Text	4000	As per the Ledger configuration
Segment9	Text	4000	As per the Ledger configuration
Segment10	Text	4000	As per the Ledger configuration
Segment11	Text	4000	As per the Ledger configuration
PayeeId	Text	256	Funder / Lessor ID
DRCR	Text (2)	2	DR = Debit CR = Credit

Field	Type	Width	Comment
Currency	Text (3)	3	International currency code
FXDate	Date	10	Foreign Exchange Transaction Date
Amount	Number	22,6	Due payment amount
Comments	Text	4000	Schedule number
LedgerEntryLineId	Text	100	LeaseAccelerator key identifier

DUE PAYMENTS EXPORT PARAMETERS:

Regardless of the integration method selected (RESTful APIs or file-based FTP) or as a scheduled report, LeaseAccelerator allows users to specify several parameters to generate Due Payments:

Monthly due payments

Parameter	Description	Required / Optional	Comment
As At	Report date	Required	
Starting Fiscal Year	Users can select specific year or relative values: ■ Current fiscal year ■ Prior fiscal year	Required	For SFTP integration, select relative value.
Starting Fiscal Month	Users can specify specific month / quarter or relative month: ■ Prior month ■ Current month ■ Prior quarter ■ Current quarter	Required	For SFTP integration, select relative value.
# of Months of Lease Expense to Transfer	Number of months	Required	At least one. For integration input one.
Level of Detail	User can select details level: ■ Portfolio level ■ Schedule level ■ Asset level	Required	For integration, only choose Portfolio or Schedule level. Selecting Asset level will generate huge file.
Schedule #	If entered, only ledger entries for this schedule will be generated	Optional	Normally left black

Parameter	Description	Required / Optional	Comment
Entity	Select Entity from a dropdown list	Optional	
Cost Center	If entered, only ledger entries for this Cost Center will be generated	Optional	Normally left black
Business Unit	If entered, only ledger entries for this Business unit will be generated	Optional	Normally left black
Country	If entered, only ledger entries for this Country will be generated	Optional	Normally left black
Lessee	If entered, only ledger entries for this Lessee will be generated	Optional	Normally left black
Set of Books	A drop down for user to a ledger.	Required	
Show deals denominated in	If entered, only ledger entries for this currency will be generated	Optional	Normally left black

Weekly Payables Export:

Parameter	Description	Required / Optional	Comment
As At	Report date	Required	
Reporting Period:	Select from a dropdown list: ■ Current week ■ Prior week	Required	
Day of the Week:	Select day of week from a dropdown list (from Sunday to Saturday)	Required	
Level of Detail	User can select details level: ■ Portfolio level ■ Schedule level	Required	
Schedule #	If entered, only ledger entries for this schedule will be generated	Optional	Normally left black

Parameter	Description	Required / Optional	Comment
Entity	Select Entity from a dropdown list	Optional	
Cost Center	If entered, only ledger entries for this Cost Center will be generated	Optional	Normally left black
Business Unit	If entered, only ledger entries for this Business unit will be generated	Optional	Normally left black
Country	If entered, only ledger entries for this Country will be generated	Optional	Normally left black
Lessee	If entered, only ledger entries for this Lessee will be generated	Optional	Normally left black
Calendar Type:	Select from a dropdown list ▪ Gregorian calendar ▪ Fiscal calendar	Required	
Set of Books	A drop down for user to a ledger	Required	
Show deals denominated in	If entered, only ledger entries for this currency will be generated	Optional	Normally left black

Actual payments from ERP to LeaseAccelerator: AP-C2

When integrating LeaseAccelerator with accounts payable, there are two scenarios that determine the data that LeaseAccelerator expects in the AP-C2 connection:

- Allocating actual disbursement to corresponding due payment
- Allocating invoices as due payments

Actual disbursements allocation

For the Lessor Invoice Matched to Due Payments model, Due Payments Only model, and Lessor Invoice Only model, the actual payments in accounts payable are compared to the due payments as part of the AP-C2 connection line in the diagrammed process for each model.

Upon actual payment of lessor's invoice, LeaseAccelerator expects to receive a file of these actual payments carrying key identifiers so that it can allocate the actual payments to the due payments on the most granular level. The accounts payable actual disbursements allocation should contain the following data:

Field	Type (Format)	Can be Null	Comment
ID	Text	No	
DatePaid	Date	No	Format (mm/dd/yyyy)
Currency	Text	No	Currency code
Amount	Decimal	No	Paid amount in the Currency
ReferenceNumber	Text	No	Payment reference in accounts payable in the ERP
PaymentReferenceID	Text	Yes	A reference for each entry generated by LeaseAccelerator
PO_Number	Text	Yes	Purchase Order Number (only used if PaymentReferenceID is null).
Payment_Due_Date	Date	Yes	(mm/dd/yyyy) Payment Due Date

Invoices as actual payments allocation AP-C2

For the Lessor Invoices as Payments model, the invoices are allocated to due payments as part of the AP-C2 connection line in the diagrammed process for this model.

LeaseAccelerator expects to receive a file of accounted invoices from accounts payable with required columns (key identifiers) and other informative columns. LeaseAccelerator matches the accounts payable invoices to LeaseAccelerator due payment using the key identifiers.

Field	Type (Format)	Allowed to be Null	Comment
Currency	Text	No	Currency code
Amount	Number (two decimals)	No	Currency Precision followed. Some currencies do not have any decimals.
PO_Number	Text	Yes	Purchase Order Number or Lease Id for Real Estate Invoices
Payment_Due_Date	Date (MM/DD/YYYY)	Yes	Payment Due Date. This is Invoice Date from AP. LeaseAccelerator uses only Month and Year (MM/YYYY for matching).
PassThrough1	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough2	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough3	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough4	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough5	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.

Field	Type (Format)	Allowed to be Null	Comment
PassThrough6	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough7	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough8	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough9	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.
PassThrough10	Text	Yes	This is sent to LeaseAccelerator to use in Exception Report.

LeaseAccelerator Application of Actual Payments to Due Payments

Based on the configuration, selected standards, information, lease terms and conditions, and allocations stored in LeaseAccelerator, LeaseAccelerator generates entries to accounts payable in the ERP for assumed payments. These payments can be generated on an asset, schedule, or portfolio level.

These due payments affect the accounts payable clearing account. When sent from LeaseAccelerator to accounts payable in the ERP, the entries include a unique payment ID.

Upon actual payment of a lessors' invoices, LeaseAccelerator expects to receive a file of these actual payments carrying key identifiers so that it can allocate the actual payments to the due payments on the most granular level.

If LeaseAccelerator key identifier (LedgerEntryLineID) is included, LeaseAccelerator will allocate the actual payment to the corresponding due payment.

Note: LedgerEntryLineID can be used multiple times and the amount can be Positive or Negative (for correction, reverse payments, or credit notes).

There are two ways to allocated actual payments to due payments in LeaseAccelerator:

1. **Using PAYMENTREFERENCE_ID**

When LeaseAccelerator exports the Due Payments, each record will have PaymentReferenceID (some versions called LedgerEntryLineID). This is the key identifier LeaseAccelerator needs when the client sends back the actual disbursements so that LeaseAccelerator can match these actual payments to their corresponding due payment.

In this case, the PaymentReferenceID will ignore all other parameters sent (Currency, PO number, etc.) and force LeaseAccelerator to accept the actual payment into the due payment even if the amount in not matching. (You can do partial payment, full payment or overpayment).

The user can also send negative amounts (in the case of correcting errors or refunds).

2. **Using other keys to allocate payment (auto-matching)**

If the key identifier (**PaymentReferenceID**) is null, LeaseAccelerator will examine the following keys:

a. Currency

b. Payment Due Date (month and year only)

c. Purchase Order Number

d. Lessor_Reference_Id

LeaseAccelerator will first search on POs (within the due payment period and assuming the currency matches), and then:

- If one due payment is fetched, then LeaseAccelerator will apply the actual disbursements to it
- If there are no records returned, then LeaseAccelerator will assume it is a non-lease payment and ignore it
- If multiple records are returned, LeaseAccelerator will use the secondary key which is “Lessor reference Id” that should result in one due payment (otherwise the matching will fail).

LeaseAccelerator will automatically apply the actual payments to the due payments only when there is an EXACT match of all the keys and the amount.

In addition to the above four mandatory columns, clients may also send 10 “PassThrough” text fields, which are used while sending the exception report.

Any mismatching records are rejected and added to the exception report along with other informative columns.

LeaseAccelerator exception report:

When uploading actual payments (or invoices), LeaseAccelerator will match the payments based on the above rules and generate the “Exception Report” so users can identify the matched and unmatched entries.

Users can fix errors and reload the same file again and so on.

Field	Type (Format)	Comment	Upload/Result	Matching Key
STATUS	Text	Matching result, could be: ▪ Matched ▪ Unable to locate PO ▪ Unable to find date for PO ▪ Unable to match amount ▪ Entry already marked as paid ▪ Unable to match currency ▪ Multiple POs found ▪ Matched on LedgerEntryLineId ▪ Invalid LedgerEntryLineId	Result	
PO NUMBER	Text	Purchase Order number – must be sent if LedgerEntryLineId is not supplied	Upload	Yes
DUE PAYMENT DATE	Text	Only month and year will be used - must be sent if LedgerEntryLineId is not supplied	Upload	Yes (month and year)
PAID AMOUNT (invoice amount)	Number	Original due payment amount	Upload	Yes
CURRENCY	Text	Due payment currency	Upload	Yes
LeaseAccelerator AMOUNT	Number	LeaseAccelerator Original Due payment amount	Result	
LeaseAccelerator BALANCE	Number	LeaseAccelerator Remaining unpaid balance	Result	Yes
PASSTHRU1	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU2	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU3	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU4	Text	Payment information – to be passed through from upload to result.	Upload	

Field	Type (Format)	Comment	Upload/Result	Matching Key
PASSTHRU5	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU6	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU7	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU8	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU9	Text	Payment information – to be passed through from upload to result.	Upload	
PASSTHRU10	Text	Payment information – to be passed through from upload to result.	Upload	
LedgerEntryLineId	Text	Due payment key identifier – if uploaded, LA will ignore other keys	Upload	Yes

FX Rates Integration

LeaseAccelerator manages leases and transactions in all known currencies. It maintains a database for currency exchange (FX) rates for all currency pairs needed by the client.

Business architecture

LeaseAccelerator allows each ledger to have one functional currency and one reporting currency. Each currency pair can have two different FX rates:

- **Spot.** A spot exchange rate is the price to exchange one currency for another for immediate delivery, which is normally used to reflect daily changes in FX rates.
- **Weighted Average.** A weekly, monthly, quarterly, or yearly average exchange rate, which is used to reflect an accounting rate that is less fluctuating. Some companies call it *"corporate rate"* which can be fixed for a week, a month, or across a full accounting period or fiscal year.

FX rates have an effective date. The rates remain effective until LeaseAccelerator receives rates with a later effective date. LeaseAccelerator adds the new rates and marks them as current; LeaseAccelerator does not overwrite older rates. If LeaseAccelerator receives FX rates with an effective date equal to the existing effective date, LeaseAccelerator overwrites the existing FX rates.

To support LeaseAccelerator's use of FX rates, the client is expected to do the following:

- Send new FX rates as soon as they change.
- Send all rates including inverse rates (However, LeaseAccelerator will calculate any missing inverse rates).
- Send all rates with up to **12** decimals. LeaseAccelerator truncates any further digits.
- Only use ISO international standard 3-digit codes for currencies.
- Either the "To currency" or the "From Currency" must be the base currency.

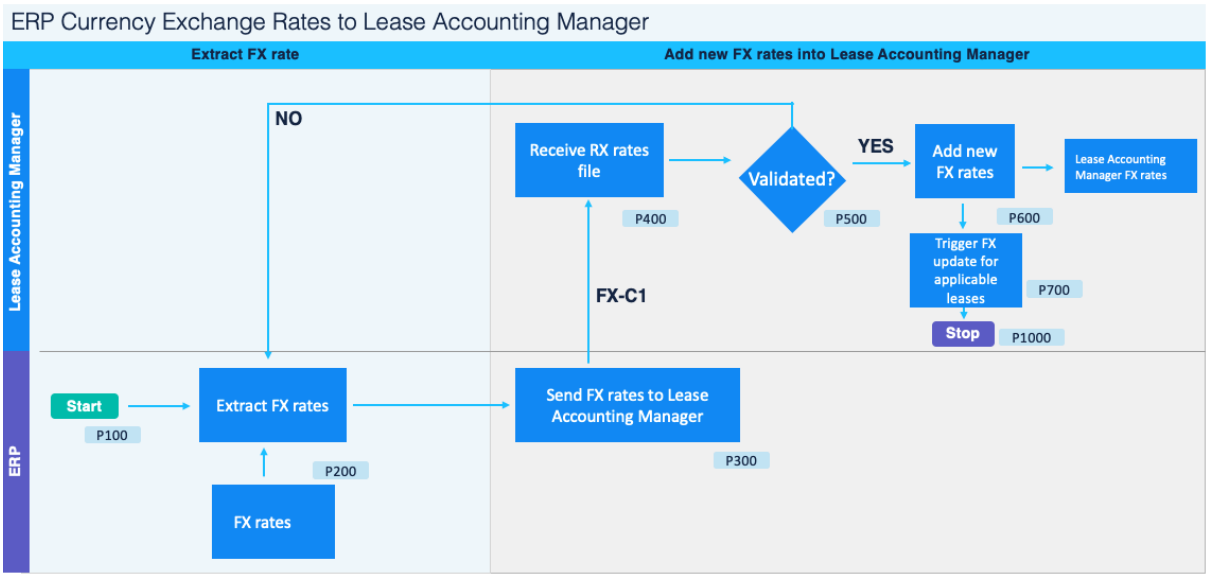
Timing

Once a lease contract (deal or schedule) is "Booked", LeaseAccelerator's accounting engine will automatically run and generate all the accounting entries and calculations for the lifetime of the lease. Once the accounting engine completes the calculations, LeaseAccelerator "sweeps" all the resulting entries into the "Reporting Engine."

Consequently, updating FX rates will trigger a "sweep" for all lease contracts that have a currency different from the corresponding ledger's currency (functional currency) or reporting currency to apply the new FX rates. The re-sweep affects all deals with ledger postings on or after the effective date of the FX rate(s) loaded.

The reporting engine may be suspended during the "sweep" duration. The sweep duration will vary depending on the number, complexity, and size of these leases. Hence, users are advised to consider the sweep time when choosing the timing and frequency of FX updates.

Integration process



Process Step ID	Process Step Description	Phase
P100	Start	Extract FX rates
P200	Extract FX Rates	Extract FX rates
P300	Send FX Rates to LeaseAccelerator	Add new FX rates into LeaseAccelerator
P400	Receive FX Rates File	Add new FX rates into LeaseAccelerator
P500	Validated?	Add new FX rates into LeaseAccelerator
P600	Add New FX Rates	Add new FX rates into LeaseAccelerator
P700	Trigger FX update for applicable leases	Add new FX rates into LeaseAccelerator
P1000	Stop	Add new FX rates into LeaseAccelerator

P100: Start

This process must be triggered daily to update FX rates. The rates remain in effect until LeaseAccelerator receives rates with a later effective date.

P200: Extract FX rates

The client's integration engine automatically triggers a process to extract the latest FX rates.

P300: Send FX rates to LeaseAccelerator

Method	Process	Next Step
RESTful API	The client's integration engine calls LeaseAccelerator FX API (ImportExchangeRates) with proper request data and payload. LeaseAccelerator validates and imports the payload. LeaseAccelerator acknowledges the request with an OK response.	P500

Method	Process	Next Step
File-based	The client's integration engine places the file into the client's SFTP folder as per the naming convention agreed. LeaseAccelerator file watcher will detect the file and ingest it, then delete it form the client's SFTP folder.	P400

P400: Receive FX rates file

LeaseAccelerator's integration engine automatically detects the FX file (Note that FX rates can also be uploaded manually using the Bulk Import functionality in the LeaseAccelerator application).

P500: Validated?

LeaseAccelerator inspects the file for errors and potentially invalid entries for currency codes, rate types, dates, and values.

Method	Process	Next Step
RESTful API	If the file contains any errors, LeaseAccelerator rejects it and sends error status in the API Response. The FX rates must be corrected, and the API called again. (P200)	P600
	Otherwise, LeaseAccelerator acknowledges the request with an OK response.	
File-based	If the file contains any errors, LeaseAccelerator rejects it and sends an email alert to the designated person. (P200).	P600

P600: Add new FX rates

LeaseAccelerator adds the new FX rates into its database and sends a message to the designated user.

If the effective date equals an existing effective date, LeaseAccelerator will overwrite the FX rates for that date.

P600: Trigger FX update for applicable leases

Updating FX rates will trigger a "sweep" for all lease contracts that have a currency different from the corresponding ledger's currency (functional currency) or reporting currency to apply the new FX rates.

P1000: Stop

The process ends, and the client updates the proper log.

Additional use cases

Case trigger	Wrong rates were sent to LeaseAccelerator.
Scenario	A file with wrong data was sent to LeaseAccelerator and was uploaded.
Steps and Expected Outcome: The client generates another file with the correct data and re-uploads it. LeaseAccelerator overwrites the old data.	

Case trigger	A reverse exchange rate for a currency pair was missing.
Scenario	The client sends the exchange rate from one currency to the another.
Steps and Expected Outcome: LeaseAccelerator automatically calculates the missing inverse value.	

Integration data

During the integration process, LeaseAccelerator receives data from the FX file as shown in the FX-C1 connection in the integration process diagram on page 81.

FX rates from ERP to LeaseAccelerator

LeaseAccelerator expects the data shown in the table from the FX rates file.

Field	Type (Format)	Allowed to be Null	Comment
To Currency	3-character string	No	Target currency code from standard international currency codes and not the same as From Currency
From Currency	3-character string	No	Source currency code from standard international currency code
Effective Date	Date (mm/dd/yyyy)	No	Effective Date
Rate	Decimal	No	Greater than 0.0 Up to 12 decimal places
Rate Type	Text	No	Spot or Weighted Average
Source	String	Yes	Source of Rate - free text to indicate source system

For each currency pair, the client sends two records for the exchange rate and the inverse rate. However, if one is missing, LeaseAccelerator automatically creates inverse rates.

For file-based integration, the column (field) order does not matter. However, the column names must match, even with the spaces. So, it should be something like:

Effective Date,From Currency,To Currency,Rate,Rate Type,Source

Balance Sheet Account Balances Integration

LeaseAccelerator integrates with Enhanced Finance Controls and Automation (EFCA) software solutions, that require Balance Sheet account balances, such as BlackLine, that are used for reconciling subledger balances with general ledger balances.

Business architecture

Immediately after month-end closing, all balance sheet account balances for primary (i.e., functional) ledgers must be transferred from LeaseAccelerator to BlackLine. LeaseAccelerator can be configured to automatically trigger the generation of a BlackLine export file in .txt format which is then imported into BlackLine.

If there are no YTD balances for an account, a zero (0.00) balance is sent.

While calculating balances, LeaseAccelerator identifies all the balance sheet accounts dynamically. If any new balance sheet accounts are added, then LeaseAccelerator considers those new accounts while extracting balances.

Account balances (i.e., YTD balances) are transferred in one export file with functional currencies for all ledgers. LeaseAccelerator converts all transactional currencies to functional currencies per the configuration settings in the LeaseAccelerator application.

Configuration

A LeaseAccelerator report must be set up and scheduled for LeaseAccelerator to automatically generate the BlackLine export file on a specific day of each month immediately after the month-end closing.

To configure the report, go to Reporting then Integration then Export BlackLine Balances in the LeaseAccelerator application.

Set the key parameters for the report as follows:

Parameter	Value
As at:	Report date. Choose future and past dates and LeaseAccelerator simulates the date as if it was generated at that date.
Ending Fiscal Year	Select current year (default).
Ending Fiscal Period	Select Current Period.
Starting Fiscal Year	Select the relative choice. For example, select Current Year instead of 2018.
Starting Period	Select the relative choice. For example, select Current Month instead of January.

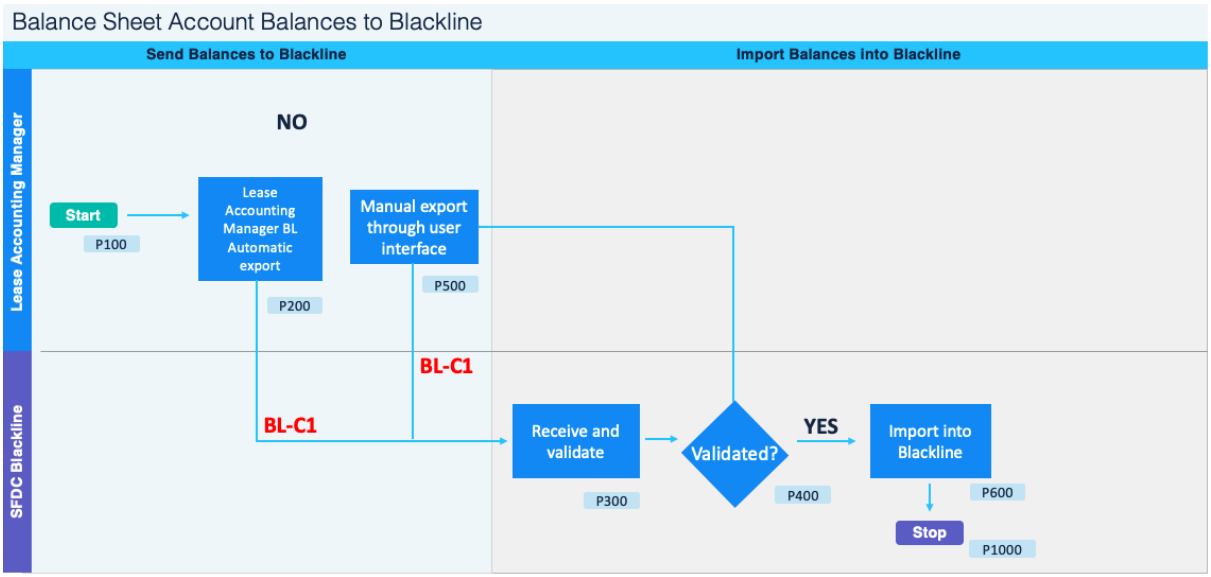
The As at, Ending Fiscal Year, and Ending Fiscal Period are required parameters.

If you do not select a relative choice for the starting parameters, LeaseAccelerator will always use the time period selected. For example, if you select January 2018 instead of current month and current year, LeaseAccelerator will send the January 2018 data in February and all other months going forward.

After entering the parameters, check the box for “Schedule this report” and enter the correct format, which is .txt, and the destination, which is the LeaseAccelerator SFTP. Designate users to receive the export file via email.

LeaseAccelerator then generates balance sheet account balances for all ledgers in functional currencies into one export file and places it into the designated location and emails it.

Integration process



Process Step ID	Process Step Description	Phase
P100	Start	Send balances to BlackLine
P200	LeaseAccelerator Triggers BL Export	Send balances to BlackLine
P300	Receive and Validate	Import balances to BlackLine
P400	Validated?	Import balances to BlackLine
P500	Manual Export in LeaseAccelerator UI	Send balances to BlackLine
P600	Import into BlackLine	Import balances to BlackLine
P10000	Stop	Import balances to BlackLine

P100: Start

This process is triggered automatically by LeaseAccelerator’s BlackLine export function as configured.

P200: Automatic export (BL-C1)

As per the configured schedule, LeaseAccelerator generates the BlackLine export file and automatically places the encrypted file into the LeaseAccelerator SFTP area. LeaseAccelerator also sends a copy of the exported file to the designated users as per the configuration.

P300: Receive and validate

BlackLine detects the new file and ingests. The BlackLine technical integration officer runs tools to validate entries in the file.

P400: Validated?

The validation tools may discover two types of errors that will prevent posting these entries into BlackLine:

- Soft errors such as ledger codes, currency code, or accounts are invalid, or a null value was received for a non-null field.
- Hard errors, which result from file corruption due to transmission or disk I/O errors.

In all cases, any invalid data in the file causes the entire file to be rejected. If there are errors, proceed to step P500.

If entries are valid, proceed to step P600.

P500: Manual export

Errors in the export file must be corrected. After correcting, the integration officer re-generates the export manually in the LeaseAccelerator application. LeaseAccelerator places the re-generated export file in the designated location and emails it to the designated users.

Return to step P300.

P600: Import

Once the file is validated, the integration officer imports the balances into BlackLine.

P1000: Stop

A log must record the success of the process.

Additional use cases

Case number	1
Case trigger	Ensure no duplicate balances can be sent to BlackLine.
Scenario	A user may – by mistake – generate an export file that has been already imported into BlackLine.
Steps and Expected Outcome: New balances overwrite the existing balances.	

Case number	2
Case trigger	Error message displayed in LeaseAccelerator during manual export process.
Scenario	When the integration officer clicks export, LeaseAccelerator displays an error message indicating that LeaseAccelerator failed to post the file to the designated location.

Steps and Expected Outcome: The integration officer should attempt to fix the failure by contacting internal IT resources or LeaseAccelerator Client Support if necessary. If the problem cannot be fixed immediately and/or the officer wishes to proceed with the integration, export the files by email or transfer to a different destination then place them manually in BlackLine import area. Then proceed with the integration process.
Warning: A control process must be in place to handle this situation.

Account balance integration data

As part of the BlackLine integration, LeaseAccelerator includes the following fields in the export file:

Field	Type (Format)	Comment
CompanyCode	Text	Company Code (Segment 1)
AccountNumber	Text	Account Number (Segment 6)
InterCompany	Text	InterCompany (Segment 2)
Ledger Name	Text	Name of the Ledger
Segment3	Null	This column is part of BlackLine template. Send Null value.
Segment4	Null	This column is part of BlackLine template. Send Null value.
Segment5	Null	This column is part of BlackLine template. Send Null value.
Key8	Null	This column is part of BlackLine template. Send Null value.
Key9	Null	This column is part of BlackLine template. Send Null value.
Key10	Null	This column is part of BlackLine template. Send Null value.
Period End Date	Date (mm/dd/yyyy)	Last day of the period
Subledger Reporting Balance	Null	This column is part of BlackLine template. Send Null value.
Subledger Functional Balance	Null	This column is part of BlackLine template. Send Null value.
Subledger Accounted Balance	Number (2 decimal places)	This is the FUNCTIONAL BALANCE. Can be Positive or Negative. Currency precision followed. Some currencies do not have any precision or decimals.

Master (Configuration) Data Integration

LeaseAccelerator classifies the configuration data (master data) into three “classes”. Members of each class are of the same “nature” so they “share” the same attributes. Each member of a certain class may have one or more “roles” when allowed. A member of a class may have a parent or its own role or another role whenever allowed.

For example, the “People class” is used to store information of all humans dealing with LeaseAccelerator though each member may have different role(s) such as: system user, asset user, asset owner, etc., but since they are all “human” they are included under the “People” class and they share common attributes such as e-mail address, telephone number, etc.

1 - Companies Class:

The “Companies” class defines those entities that are part of your leasing program. Each “company” however may perform one or more role(s) or function(s) – if applicable. Companies class include lessees, funders, business entities, business units, vendors, and cost centers.

2 – People Class:

Many people participate in your leasing program and have different roles. These people may or may not be users of the system. LeaseAccelerator has a dedicated class for “people” to capture information about these people so that they can receive notifications or other information important to their function. Examples of the roles that people may play and therefore need to be configured in the system are: Asset Owners, Asset Users, Order Administrators, Analysts, Finance Approvers, etc.

3 – Addresses Class:

Addresses are specific significant locations in your organization. These addresses can be workplaces, physical location of leased equipment, a shipped to address or the physical location of real estate.

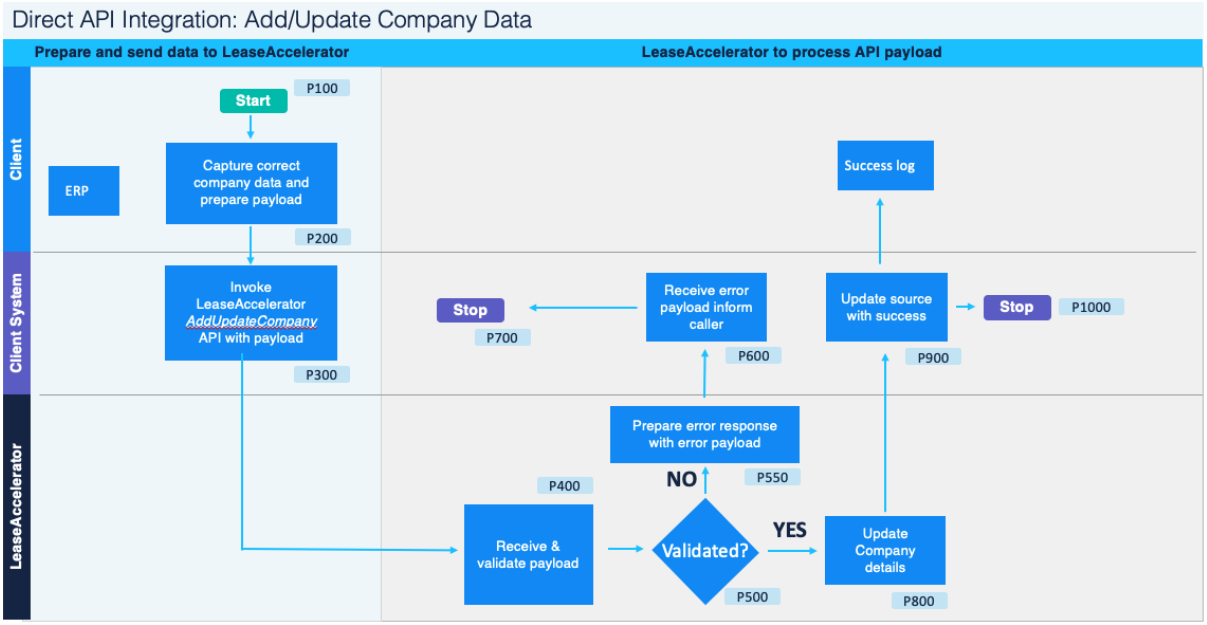
Companies class integration

The “Companies” class defines those entities that are a part of your leasing program. Each “company” however may perform one or more role(s) or function(s) – if applicable. Companies class include:

- **Lessees:** The legal party to the agreement who are leasing the equipment or renting the real estate
- **Funders:** Also referred to as lessors: they provide the cash to finance the equipment purchase through a lease agreement
- **Entity:** Organization you consider to be the Lessee (may be different than actual Lessee)
- **Business Units (SBU):** An internal reporting group
- **Vendors:** Various other parties such as the equipment manufacturer or supplier, an assignee, or an organization to whom you pay an initial direct cost to or receive a lease incentive from

All members of Companies class share the same attributes (see Data Mapping for details).

Adding and updating companies process flow



Process Step ID	Process Step Description	Phase
100	Start	Source system user
200	Capture Companies data and prepare payload	Source system user
300	Invoke LeaseAccelerator AddUpdateCompany API with payload	Middleware
400	Receive & validate payload	LeaseAccelerator
500	Valid?	LeaseAccelerator
550	Prepare error response with error payload	LeaseAccelerator
600		Middleware
700	Stop	Middleware
800	Add /Update Companies details	LeaseAccelerator
900	Update source with success	Middleware
1000	Stop	Middleware

Companies class data mapping

	Field Name	Description	Format	Req'd System Import	Req'd Acctg
1	CompanyRoleType	This is a description of the function that this company plays within your leasing program.	See Data Validation List	Y	Y

	Field Name	Description	Format	Req'd System Import	Req'd Acctg
2	CompanyName	Name of the entity which you want to use within the system. This may or may not be the official legal name of the entity. This may be a subsidiary of a larger organization.	Alphanumeric (150)	Y	Y
3	ParentCompanyName	The highest organization within a company's legal structure.	Alphanumeric (150)	N	N
4	URL	Website Address	Alphanumeric (512)	N	N
5	AddressLine1	Street Address	Alphanumeric (250)	N	N
6	AddressLine2	Additional address information such as floor or suite.	Alphanumeric (250)	N	N
7	City	City used by postal service. This may not be the more common name used in conversation.	Alphanumeric (100)	N	N
8	StateProvince	Postal Abbreviation	Alphanumeric (4)	N	N
9	Country	Name	See Data Validation List	N	N
10	PostalCode	Zip Code or other Postal code	Alphanumeric (16)	N	N
11	ContactName	First Name, Last Name of Primary Contact	Alphanumeric (150)	N	N
12	Phone	Telephone number	Numeric (32)	N	N
13	Email	Email address for POC	Alphanumeric (150)	N	N
14	LedgerCode	The code per Chart of Accounts associated with the participant company, if appropriate.	Alphanumeric (32)	N	Y

People class integration

LeaseAccelerator has a dedicated class for "people" to capture information about people who may have specific roles whether they are system users or not.

LeaseAccelerator supports the following classes of people:

- **Asset Owners:** Employees who have fiduciary responsibility for the maintenance and use of the assets. These employees are typically management level and are financially responsible for the equipment.

- **Asset Users:** Employees who have custodial responsibility for the leased assets. These employees may be management or staff level. These employees' work locations are typically at the equipment's physical location or in the same area.

- **Order Administrators:** Employees who have "procurement" responsibility for the leased assets, shepherding the administrative process in your organization to finance the acquisition of equipment and securing its delivery to the asset user.

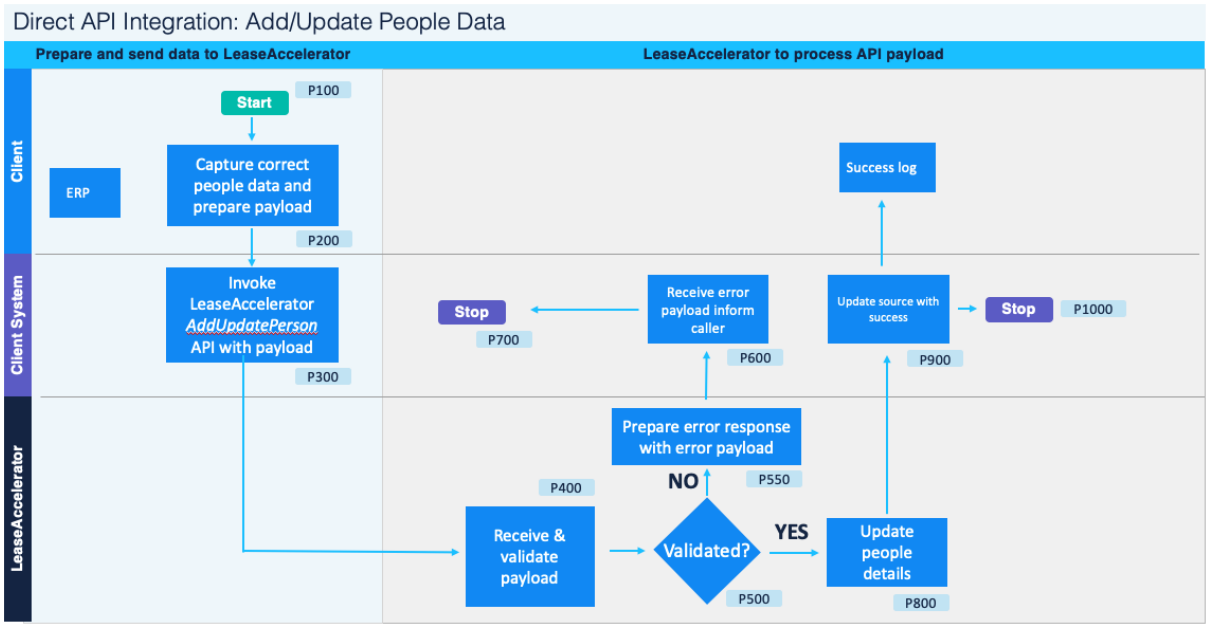
- **Analysts:** Employees responsible for generating the Lease vs. Buy.

- **Finance Approvers:** Employees typically responsible for approving the lease transaction and may be the responsible party signing the lease documents. These employees are typically management or executive level and are an escalation step for notifications regarding End-of-Term.

- **Sourcing contacts:** Employees responsible for initiating and creating an RFP, reviewing the proposals and awarding to a Funder/Lessor.

Please note that a person may be listed multiple times if their role types are different, but the company and address information for that person must be identical. Do not list a person multiple times with the same role type and different companies.

Adding and updating people process flow



Process Step ID	Process Step Description	Phase
100	Start	Source system user
200	Capture People data and prepare payload	Source system user
300	Invoke LeaseAccelerator **AddUpdatePerson** API with payload	Middleware

Process Step ID	Process Step Description	Phase
400	Receive & validate payload	LeaseAccelerator
500	Valid?	LeaseAccelerator
550	Prepare error response with error payload	LeaseAccelerator
600	Receive error payload & inform caller	Middleware
700	Stop	Middleware
800	Add /Update People details	LeaseAccelerator
900	Update source with success	Middleware
1000	Stop	Middleware

People class data mapping

People operations have no accounting effect.

#	Field Name	Description	Validation / Valid Values	Format	Required
A	Full Name	First name, Last name		Alphanumeric (150)	Y
B	Person Role Type	The function that a person plays within your leasing program.	See Data Validation List	See Data Validation List	Y
C	Company Name	Employee's Entity or Business Unit (typically governed by your company's internal method to commonly reference the organization that an employee is part of).	Must exactly match a value on Companies tab where Company Role Type = Lessee, Entity, Funder, Vendor, or SBU.	Alphanumeric (150)	Y
D	Address Line 1	Street Address	If blank, data from corresponding Company Name on Companies tab will be applied.	Alphanumeric (250)	N
E	Address Line 2	Additional address information such as floor or suite.	If blank, data from corresponding Company Name on Companies tab will be applied.	Alphanumeric (250)	N

#	Field Name	Description	Validation / Valid Values	Format	Required
F	City	City used by postal service. This may not be the more common name used in conversation.	If blank, data from corresponding Company Name on Companies tab will be applied. Required for LeaseAccelerator users. Defined on LeaseAccelerator Users tab.	Alphanumeric (100)	Y
G	State Province	Postal Abbreviation	If blank, data from corresponding Company Name on Companies tab will be applied.	Alphanumeric (4)	N
H	Country	Postal Abbreviation	If blank, data from corresponding Company Name on Companies tab will be applied. Required for LeaseAccelerator users. Defined on LeaseAccelerator Users tab.	Dropdown	Y
I	Postal Code	Zip Code or other Postal code	If blank, data from corresponding Company Name on Companies tab will be applied.	Alphanumeric (16)	N
K	Phone	Telephone number		Numeric (32)	N
L	Email*	Email address for POC	Valid email address format. Must include @xxx.	Alphanumeric (150)	Y

#	Field Name	Description	Validation / Valid Values	Format	Required
M	Is User*	Field to define whether this person is a defined user within LeaseAccelerator. By selecting Yes, this person will be provisioned as a user upon import and their username will be everything before the @ sign of their email address.	Yes or No	Dropdown	Y

Addresses class integration

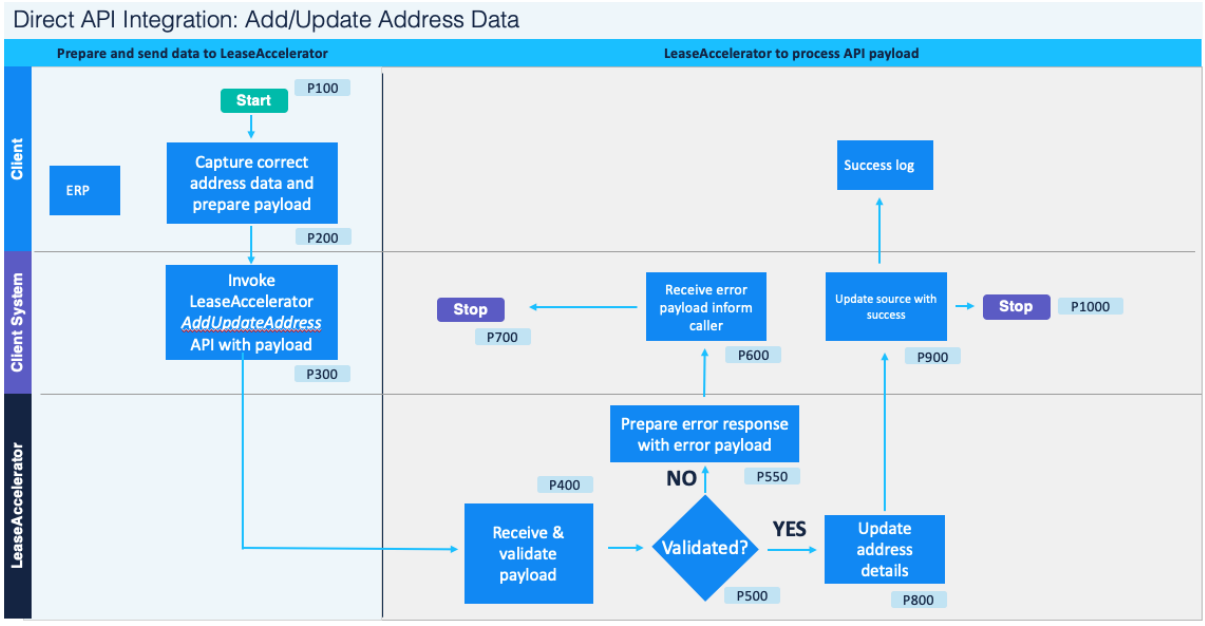
Addresses are specific significant locations in your organization. These addresses can be:

- Employees workplace
- Physical location of **equipment** or **shipped to**
- Physical location of leased **real estate**

Addresses are typically associated with a Lessee, Entity, or Business Unit. (There may be a 1: n relationship because companies may have multiple teams “sitting” at a single location.)

All members of Addresses class share the same attributes (see Configuration Data Mapping for details).

Adding and updating addresses process flow



Process Step ID	Process Step Description	Phase
100	Start	Source system user
200	Capture People data and prepare payload	Source system user
300	Invoke LeaseAccelerator **AddUpdateAddress** API with payload	Middleware
400	Receive & validate payload	LeaseAccelerator
500	Valid?	LeaseAccelerator
550	Prepare error response with error payload	LeaseAccelerator
600	Receive error payload & inform caller	Middleware
700	Stop	Middleware
800	Add /Update People details	LeaseAccelerator
900	Update source with success	Middleware
1000	Stop	Middleware

Addresses class data mapping

#	Field Name	Description	Validation / Valid Values	Format	Req'd System Import	Req'd Acctg
1	Address Role Type	The function that this location plays within your leasing program.	See Data Validation List below	Text	Y	Y
2	Company Name	Name of the location used as reference within your company where assets are located.		Alphanumeric (150)	Y	Y
3	Address Line 1	Street Address		Alphanumeric (250)	N	N
4	Address Line 2	Additional address information such as floor or suite.		Alphanumeric (250)	N	N
5	City	City used by postal service. This may not be the more common name used in conversation.		Alphanumeric (100)	Y	N
6	State Province	Postal Abbreviation	Validation to Postal Codes per USPS	Alphanumeric (4)	N	N
7	Country	Postal Abbreviation	See Data Validation List Below	Drop Down	Y	N
8	Postal Code	Zip Code or other Postal code		Alphanumeric (16)	N	N

#	Field Name	Description	Validation / Valid Values	Format	Req'd System Import	Req'd Acctg
9	Facility Code	Facility Code (ShipTo Key) is a client code associated with a specific ShipTo Address. Using a Facility Code on the PIW eliminates the need to enter any values in the ShipTo Address fields.	Unique value, code cannot exist in the system already.	Alphanumeric (100)	N	N
10	Ledger Code	The "value" used by your accounting system to identify the work location. Typically, this is one of many portions of an overall General Ledger string.		Alphanumeric (32)	N	Y

Setting FTP Folders

LeaseAccelerator allows clients to designate an SFTP folder for each ledger. This means that clients may configure all outbound integration files (ledger entries, due payments, account balances, etc.) pertaining to a specific ledger to be posted in a separate folder.

The benefits of having a separate folder for each ledger (or even a separate FTP server) depend on the IT organization structure globally and the security and data protection strategy applied at the client.

For example, a client may configure LeaseAccelerator to post all ledger entries for North America on a different server than ledger entries for EMEA and so on.

This allows segregation of data, better control, and security to protect data from possible intentional / unintentional unauthorized access.

For inbound integration, each client can only have one folder whereby Lease Accounting Manger will monitor this folder and consume any detected files then delete them.

FTP setup for outbound file transfer

LeaseAccelerator can post generated outbound integration files into a client hosted SFTP server (Secured File Transfer Protocol). LeaseAccelerator does not host nor maintain an SFTP server.

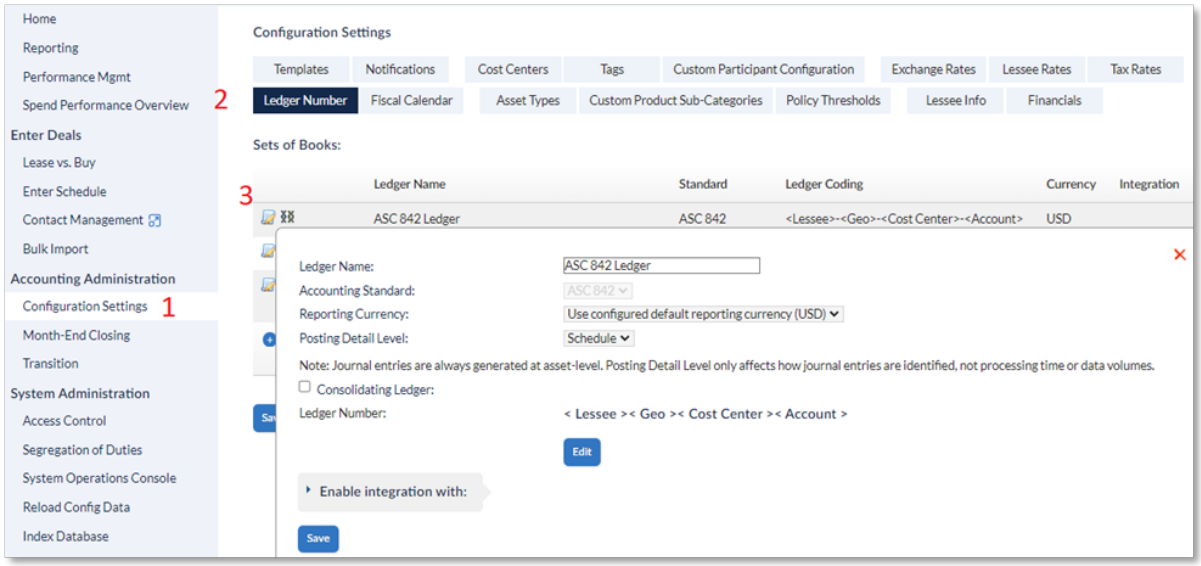
Prerequisites:

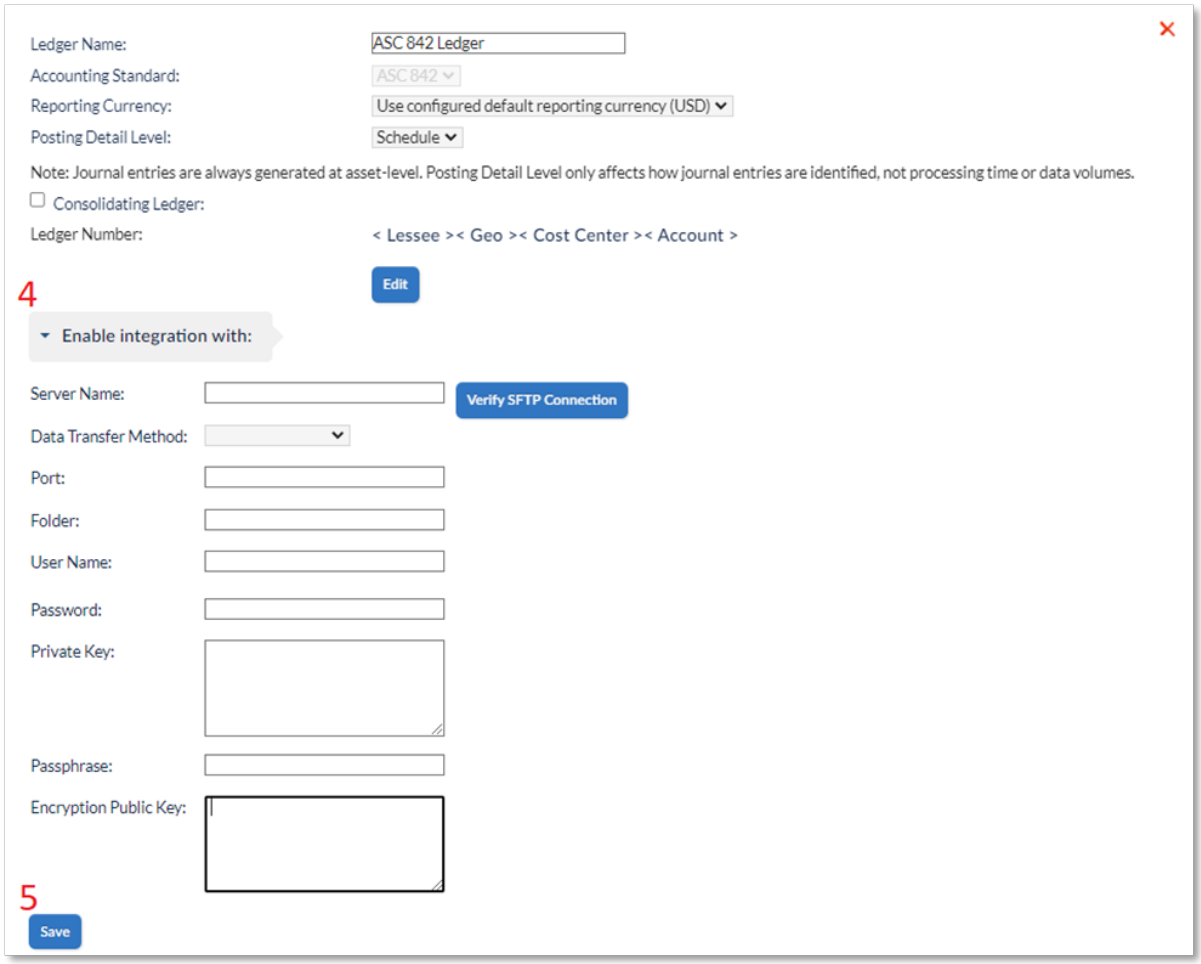
- The user must have privileges to access LeaseAccelerator system settings.
- The user must have already created the target folders on the designated SFTP servers.
- The user must have the PgP encryption key (if needed).

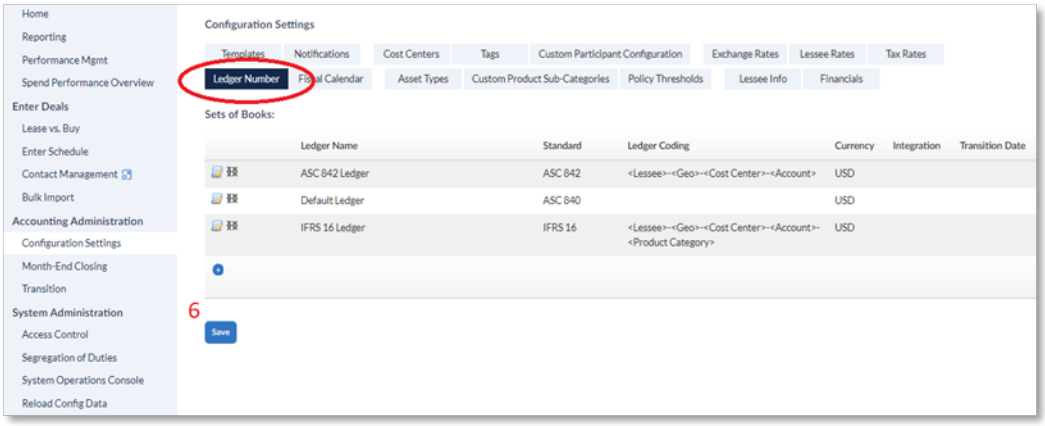
LeaseAccelerator supports PgP encryption of exported files.

These settings are to be done once for each ledger. However, users can update the information as needed.

SFTP folder configuration steps:


Home
Reporting
Performance Mgmt
Spend Performance Overview
Enter Deals
Lease vs. Buy
Enter Schedule
Contact Management
Bulk Import
Accounting Administration
Configuration Settings
Month-End Closing
Transition
System Administration
Access Control
Segregation of Duties
System Operations Console
Reload Config Data
Index Database
1
2
3
Configuration Settings
Templates
Notifications
Cost Centers
Tags
Custom Participant Configuration
Exchange Rates
Lessee Rates
Tax Rates
Ledger Number
Fiscal Calendar
Asset Types
Custom Product Sub-Categories
Policy Thresholds
Lessee Info
Financials
Sets of Books:
Ledger Name
Standard
Ledger Coding
Currency
Integration
ASC 842 Ledger
ASC 842
<Lessee>-<Geo>-<Cost Center>-<Account>
USD
Ledger Name:
ASC 842 Ledger
Accounting Standard:
ASC 842
Reporting Currency:
Use configured default reporting currency (USD)
Posting Detail Level:
Schedule
Note: Journal entries are always generated at asset-level. Posting Detail Level only affects how journal entries are identified, not processing time or data volumes.
Consolidating Ledger:
Ledger Number:
< Lessee >< Geo >< Cost Center >< Account >
Edit
Enable integration with:
Save


Ledger Name:
ASC 842 Ledger
Accounting Standard:
ASC 842
Reporting Currency:
Use configured default reporting currency (USD)
Posting Detail Level:
Schedule
Note: Journal entries are always generated at asset-level. Posting Detail Level only affects how journal entries are identified, not processing time or data volumes.
Consolidating Ledger:
Ledger Number:
< Lessee >< Geo >< Cost Center >< Account >
Edit
4
Enable integration with:
Server Name:
Verify SFTP Connection
Data Transfer Method:
Port:
Folder:
User Name:
Password:
Private Key:
Passphrase:
Encryption Public Key:
5
Save



1. Go to “settings” (the top right corner).
2. Select “Ledger Number.” LeaseAccelerator will display a list of defined ledgers.
3. On the list of ledgers, click the button to the left of the selected ledger.
4. Choose “Enable Integration with”
5. Input SFTP server and folder data.
6. Input credentials (either ID and Password) or a Private Key and a pass phrase.
7. PgP (**P**retty **G**ood **P**rivacy) encryption keys (if needed).
8. Once you enter the server, user, private authentication key, public encryption key and folder, you should be able to test the transfer. On the Settings -> Ledger Number screen you have to hit the “Save” button for the Ledger and then the “Save” button on the Ledger Number screen to save all changes.

Note: The client generates the public/private key pair, configures their FTP server to associate the public key with a user, and then enters the private key in LeaseAccelerator -> Settings -> Ledger Number screen.

Setting Outbound File Scheduler

For outbound integration files such as ledger entries and AP due payments, users can configure LeaseAccelerator to automatically generate the export files and either send them by email or post them into the FTP folder configured in the “SETTING FTP FOLDERS”.

For example, for ledger entries, if you use a file-based method, you can set the scheduler in LeaseAccelerator (Reporting -> Integration) so that LeaseAccelerator will automatically generate the file and post it in the SFTP pre-configured folder.