sage Partner Summit

API Integration Considerations

Louis Sterio

Contents

- API Overview
- Developer Website
- Security
- LOS and Throttling
- Integration Optimization
- Best Practices

Application Programming Interface (API)

- Functions you can execute
- Primarily XML-based
- Not a subscription in Sage Intacct
- Has “clients” that use it

```
<create>
    <VENDOR>
        <VENDORID>V1234</VENDORID>
        <NAME>Intacct Corp</NAME>
        <DISPLAYCONTACT>
            <PRINTAS>Intacct Corporation</PRINTAS>
        </DISPLAYCONTACT>
    </VENDOR>
</create>
```

```
<update>
    <EMPLOYEE>
        <RECORDNO>12</RECORDNO>
        <TITLE>CEO</TITLE>
    </EMPLOYEE>
</update>
```

```
<readByQuery>
    <object>APBILL</object>
    <fields>*</fields>
    <query></query>
    <pagesize>100</pagesize>
</readByQuery>
```

API Clients

Web Services

- XML gateway
- Requires Web Services Developer license

Platform Services

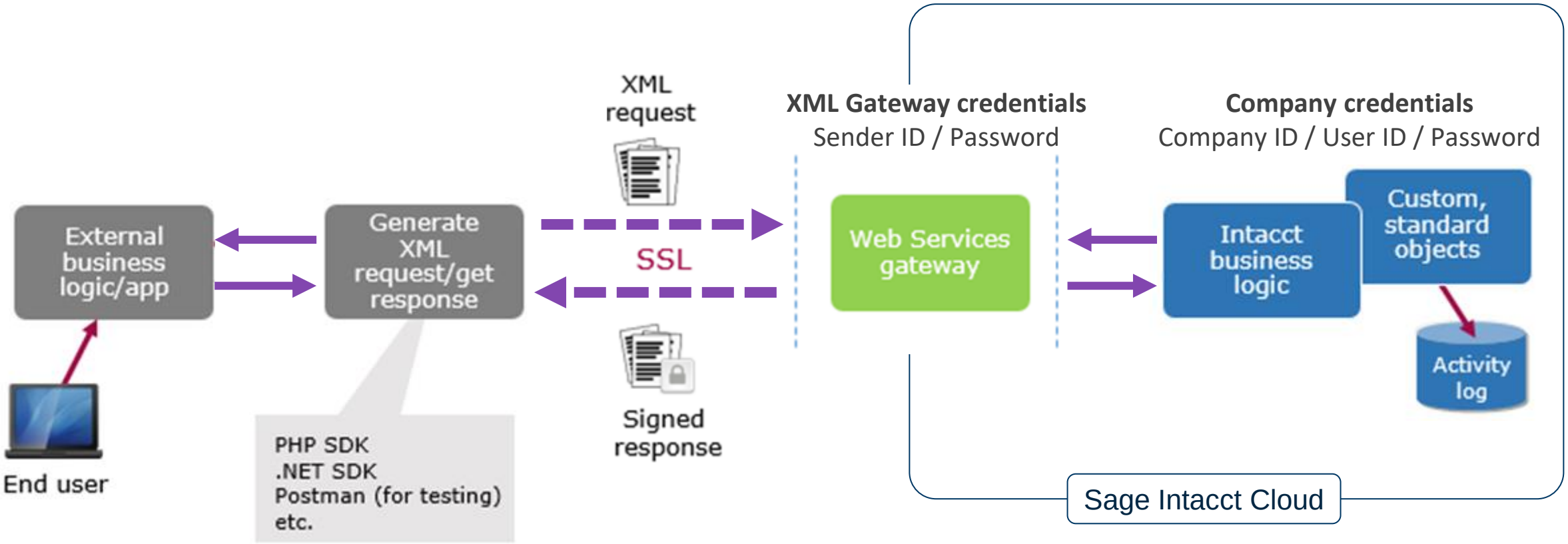
- AJAX gateway
- Custom object trigger
- Requires Platform Services Developer license

Customization Services

- Standard object Smart Event

CSV Imports

Web Services API Call



Web Service Licensing

Intacct Web Services

- Entitlement to subscribe to and use web services

Intacct Web Services Developer

- Entitlement to create API based solutions

Includes a sender ID / password

Required for developing integrations

Not required by the customer for MPP integrations (MPP integrations use the sender ID of the MPP)

Web Services

Wrapper of API functions

Subscription in Sage Intacct

XML gateway entry point

External facing

- https://api.intacct.com/ia/xml/xmlgw.phtml

Protected by sender and user credentials

Two versions: v2.1 and v3.0

Not REST or SOAP

```
<request>
    <control>
        <senderid>{{sender_id}}</senderid>
        <password>{{sender_password}}</password>
        <controlid>{{$timestamp}}</controlid>
        <uniqueid>false</uniqueid>
        <dtdversion>3.0</dtdversion>
        <includewhitespace>false</includewhitespace>
    </control>
    <operation>
        <authentication>
            <sessionid>{{temp_session_id}}</sessionid>
        </authentication>
        <content>
            <function controlid="{{$guid}}">
                <readByQuery>
                    <object>VENDOR</object>
                    <fields>*</fields>
                    <query></query>
                    <pagesize>100</pagesize>
                </readByQuery>
            </function>
        </content>
    </operation>
</request>
```

API function

Throttling and LOS

Throttling

Web Services XML Gateway

- Concurrent requests per company

Platform AJAX Gateway

- Based on UI throttle for user

Offline processes

- Smart Events/Triggers
- CSV Imports
- Offline Reports
- DDS

Premium Level of Service

Customers who require

- A high volume of API transactions or Premium operations capacity

Enables more concurrent use in the form of

- Interactive users
- API calls
- Offline processes

May also improve speed of system

Dedicated/reserved queues

- Offline jobs
- Offline reports

Higher API concurrency

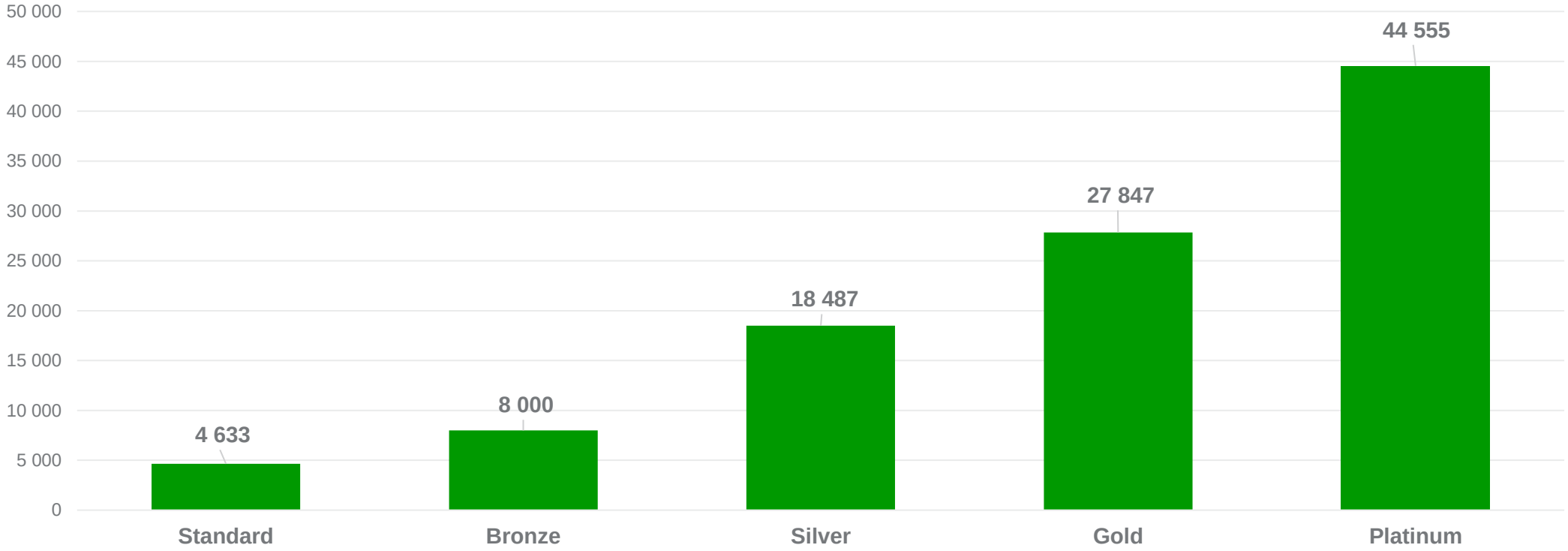
Premium Level of Service Offerings

LOS Level	Designed for Monthly TXN Volumes up to	Reserved Queues	API Concurrency
Standard	25,000	-	1
Bronze	250,000	-	2
Silver	1,000,000	2 Offline Jobs 1 Offline Reports	5
Gold	5,000,000	4 Offline Jobs 2 Offline Reports	10
Platinum	Unlimited	6 Offline Jobs 3 Offline Reports	20

Sample Throughput

Activity: Sales Invoice with A/R and GL Posting

Estimated Throughput Per Hour



- *Daily posting summaries*
- *API posts 50 invoices per request with transactions disabled*
- *Requires using API concurrency allowances*

Security Considerations

Why Discuss Security?

Create and maintain the integrity of Sage Intacct and our customer's brand image

- Loyalty, reputation, preference, and avoid brand rejection

Customers ask

- Requests for information
- Contractual obligations

Breaches are expensive and embarrassing

- ~$158/record *

Industry regulations require it

- SSAE 18, Sarbanes Oxley (for public companies)
- EU Privacy Shield, GDPR, HIPAA, PCI-DSS

Because Sage Intacct is a financial application

Sage Intacct Secures Core Components

Our Responsibility

- Secure data center
- Strong network segmentation
- Advanced firewalls
- Secure servers
- Application security
- Enhanced security monitoring
- Encryption
- Audit and compliance (SOC, GDPR, PCI)
- Disaster recovery/backups

Security Options Within Sage Intacct

Inactivity and session timeouts

Password and sign-in options

IP address filtering

- Allow access from specific IP addresses

Single sign-on (SSO)

- Integrate with an authentication provider
- One password for access to multiple services

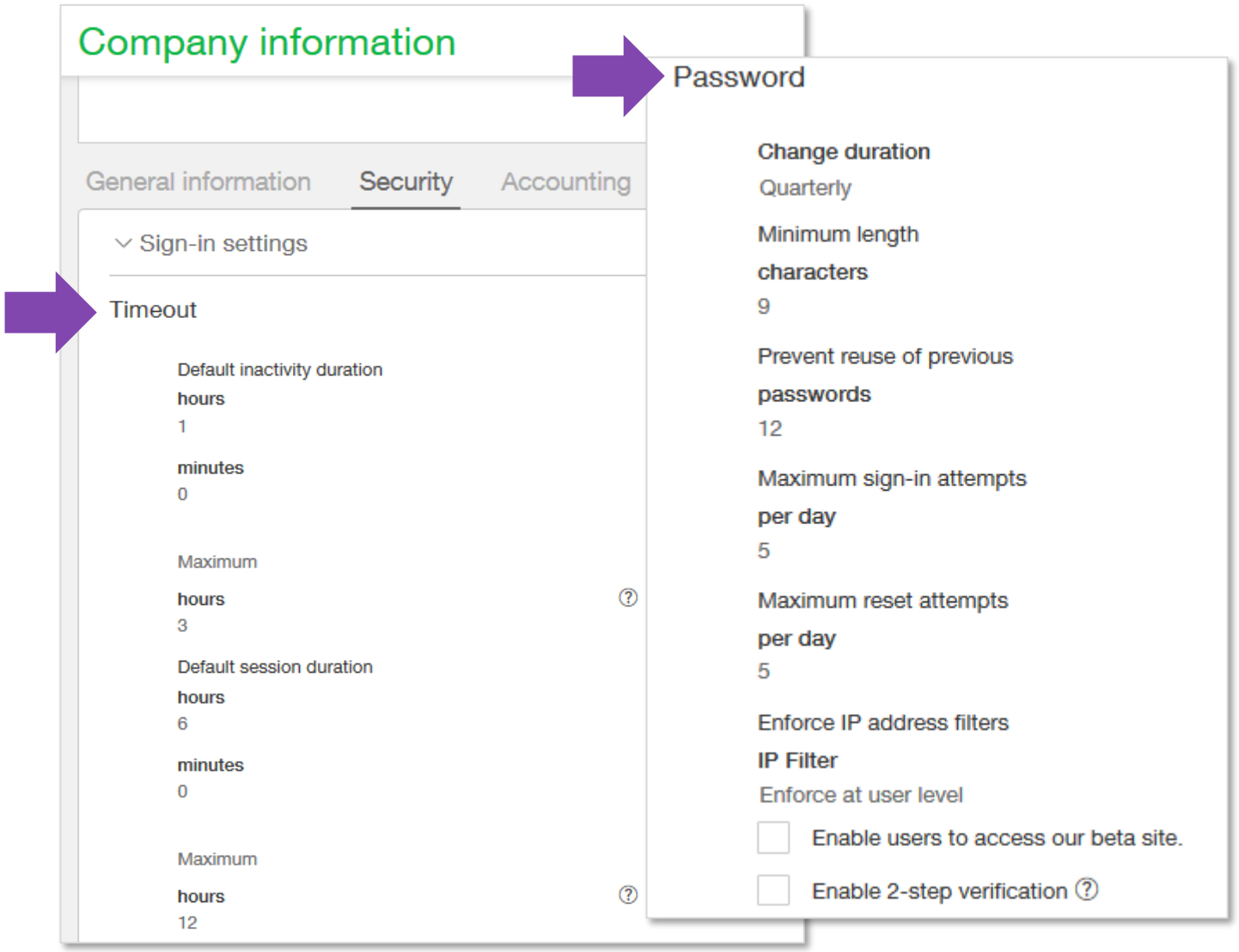
Two-step verification

- Remove trusted device option for certain users

Roles and permissions

- Limit user activity and visibility depending on their role within the organization

Configure Security Options



Best Practice Recommendations

Inactivity and Session Timeout

Inactivity duration–“empty chair” security

- Invalidates user’s session if they are idle for a period of time

Session duration–overall session security

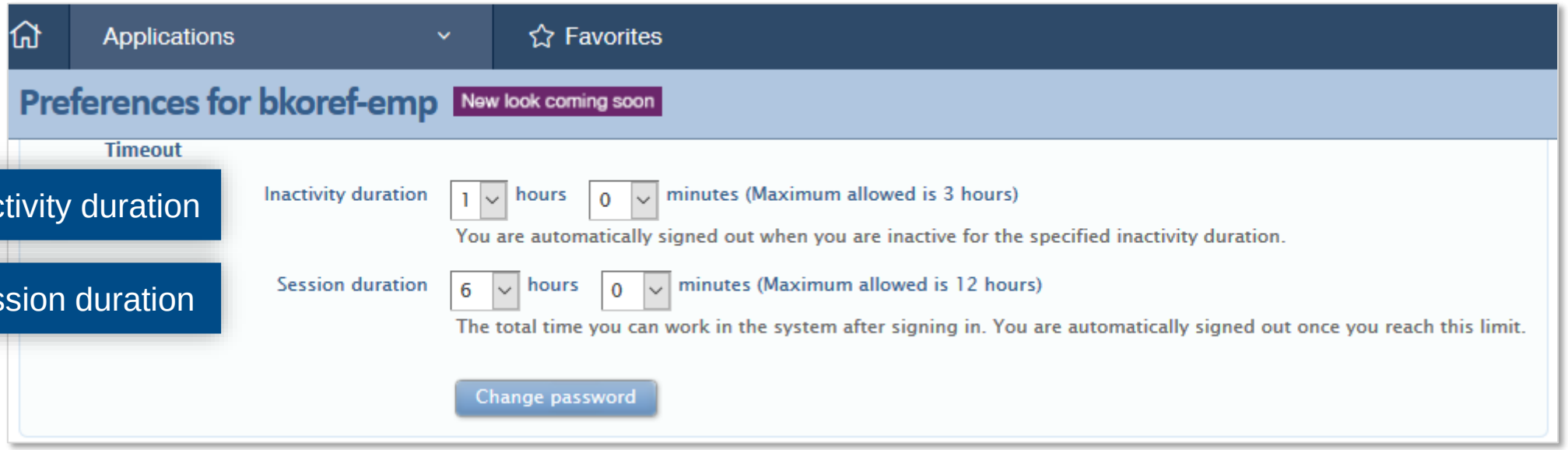
- Invalidates a user’s session after a set period of time
- Overall authenticated session duration
- Forces user to periodically re-authenticate
- Session is invalidated even if the user is active (not idle)

User Preferences for Timeouts

User can change preferences, cannot exceed company settings

Inactivity duration

Session duration



Two-step Verification (Minimizing Password Risk)

Requires an authentication token for access from an unauthorized device

Similar to the approach employed by banks, Amazon, Dropbox, Salesforce.com, Google, Facebook and many other online services

Authentication token obtained via text message, voice message, or authenticator app

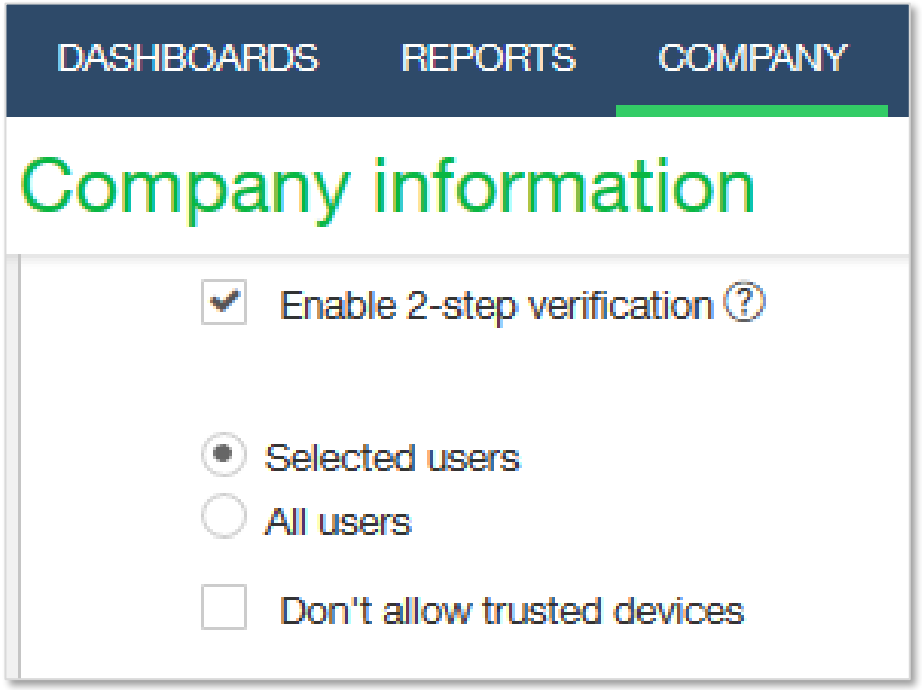
Myth – “I have to enter my token every time I login”

With two-factor authentication, mere knowledge of username and password is not sufficient to break into a user's account

Two-step Verification

Standard 2-step verification should be enabled for all consoles and production tenants

- Company > All> Configuration >Security Tab

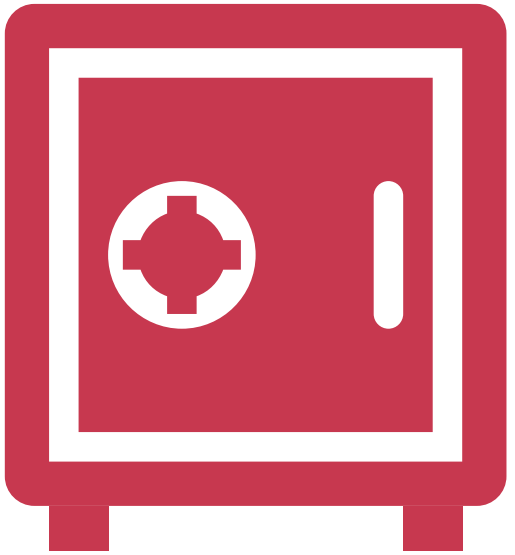


Sage Intacct Security Controls

Security Control	Purpose	Best Practice
Two-step verification	Mitigates risk of unauthorized access	Use it and don't allow trusted devices where applicable
Enforce password history and change frequency	Prevents malicious user from using previously known/compromised password	Quarterly (90 day) password change History = 12
Maximum password reset attempts	Mitigates brute force/dictionary password guessing attacks	Set to 5 or less
Maximum sign-in attempts	Mitigates risks of brute force/password guessing attacks	Set to 5 or less
Minimum password length	Mitigates risk of brute force	9 or more, especially for Admins
Default inactivity timeout	Mitigates risk associated with unattended devices	Set to 1 hours or less Admin: 15 minutes
Default session timeout	Mitigates risk of unauthorized access	5 hours or 10 hours, depending on your business environment
Enforce IP address filters	Restricts Sage Intacct access to your companies' network	Enforce for Admin's, risky users, integration (API/Web Services)
Roles and permissions	Restrict access based upon need to know	Create, use, and regularly review

Best Practices for Integration (Web Services/API)

- Utilize Restrict by IP
- Use a long and strong password
- Don’t use the same password for other accounts
- Don't log passwords, redact them if you must log
- Generate a session ID, then use that session for future requests instead of the password
- Utilize Web Services Authorization – NEW!



WS Users and WS Authorizations

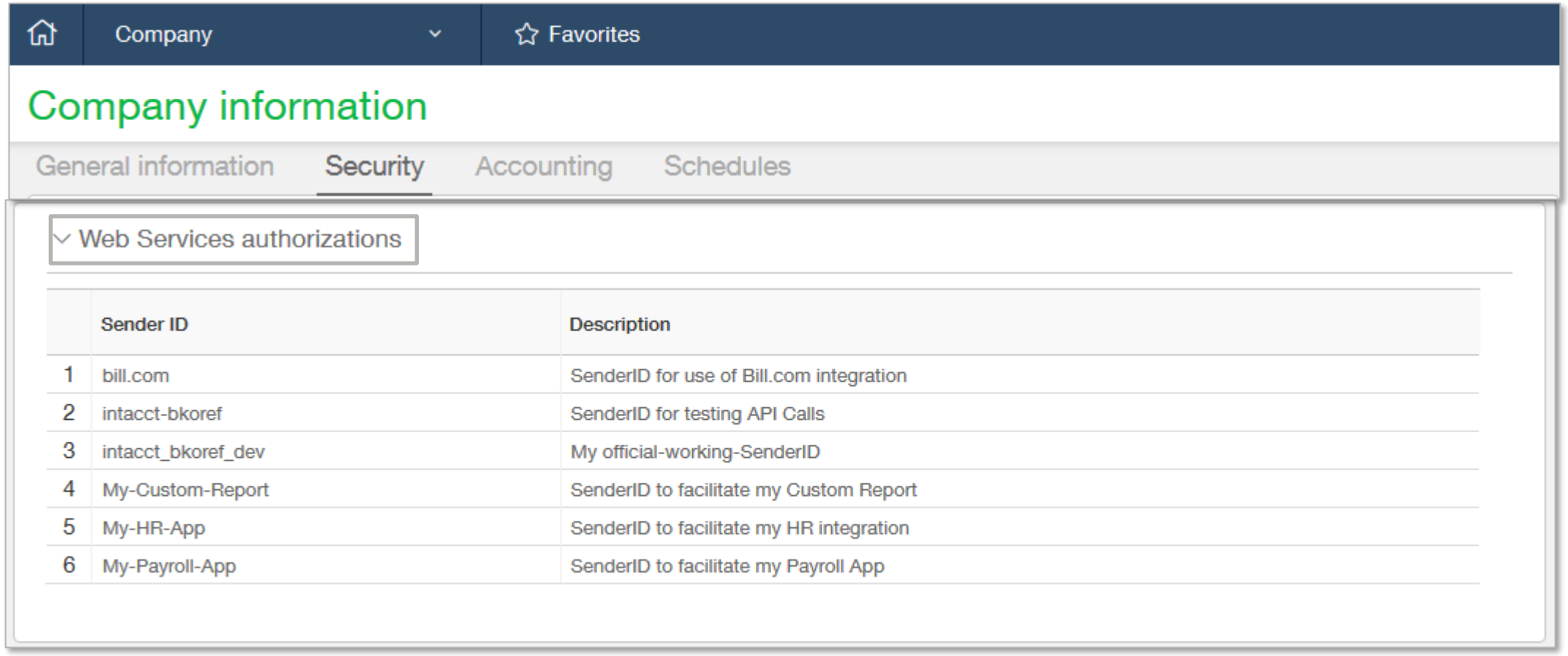
Web Services Users

- Limited to API Only
- Password remains until the admin resets it
- Cannot have single-sign on or multi-factor authentication
- Still need to have valid Sender ID

Web Services Authorizations

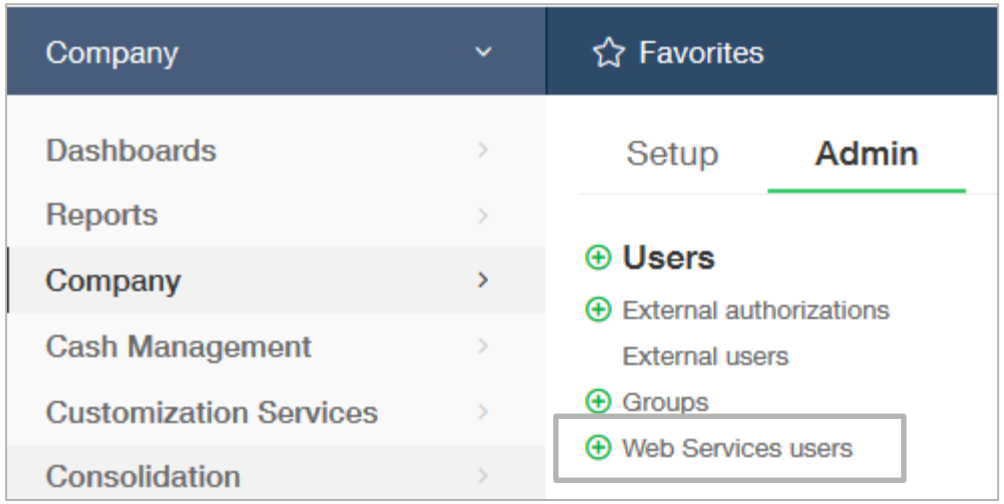
- Controls which sender IDs can and cannot make Web Services requests to your company
- If a sender ID is not on this list, any Web Services requests they make to your company will fail

Web Services Authorization



	Sender ID	Description
1	bill.com	SenderID for use of Bill.com integration
2	intacct-bkoref	SenderID for testing API Calls
3	intacct_bkoref_dev	My official-working-SenderID
4	My-Custom-Report	SenderID to facilitate my Custom Report
5	My-HR-App	SenderID to facilitate my HR integration
6	My-Payroll-App	SenderID to facilitate my Payroll App

Web Services Users



Company | Favorites

Web Services Users

Add | Done | Export

		User ID	User name	User type	Admin privileges		Permissions Report			
Edit	View	Aeinstein	Albert Einstein	Business User	Full	Subscriptions	View Permissions	Groups	Preferences	Delete
Edit	View	expensify-wsuser	expensify-wsuser	Business User	Full	Subscriptions	View Permissions	Groups	Preferences	Delete
Edit	View	reportAPI-wsuser	reportAPI-wsuser	Business User	Full	Subscriptions	View Permissions	Groups	Preferences	Delete
Edit	View	Ws-user-test	Brian Koref	Business User	Full	Subscriptions	View Permissions	Groups	Preferences	Delete

Best Practices for Integration (Web Services/API)

Create appropriate role for web services user (limit roles to the API's purpose)

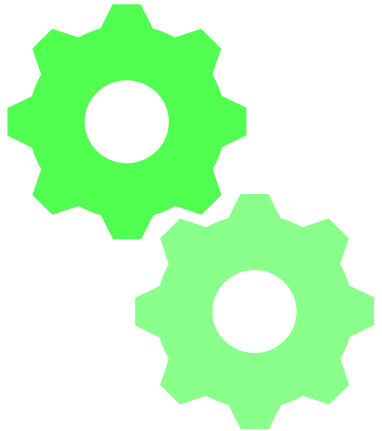
Don't tie account to a user

- Ensure the account is clearly named to identify the purpose of the account (i.e., xml_gateway-fixedassets)

Ensure the email address associated with web services account resolves to a real person

Change the password if:

- You suspect a compromise
- A knowledgeable person leaves your organization
- Your security policy or compliance requirements call for periodic change



Additional Security Best Practices

Use a unique password for your Sage Intacct account

If utilizing SSO, ensure 2-step (MFA) is enforced

Disable external access and implementation accounts when not needed

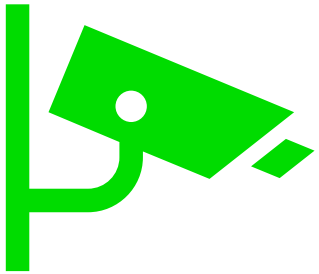
Integrations

- Understand what third parties are obtaining via an integration

Custom fields

- Data in custom fields are not encrypted

Regularly review accounts and roles



Roles: Secure Information from within Sage Intacct

Role-based vs. User-based Permissions

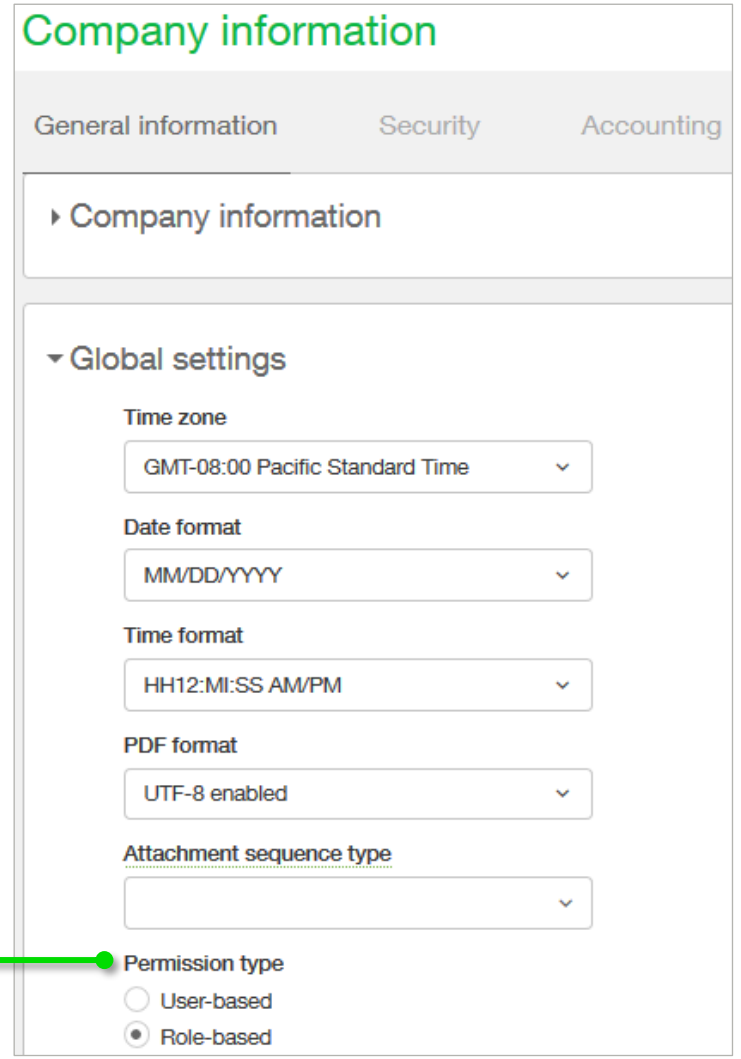
Role-based permissions are easy to manage and scale better

Role-based	User-based
• Assign permissions to roles • Assign roles to users • Reuse roles for similar users • Role “stacking”—combine multiple roles for a user	• Assign permissions to each user • Individual management of permissions can be slow and error prone
• Adjust permissions via the role—affects all users with that role	• Adjust permissions for each user
• Role assignment is copied when a user is duplicated	• User permissions are not copied when a user is duplicated

Enable Role-based Permission Management

Company > All > Configure

Previously assigned user-based roles will convert to system generated roles



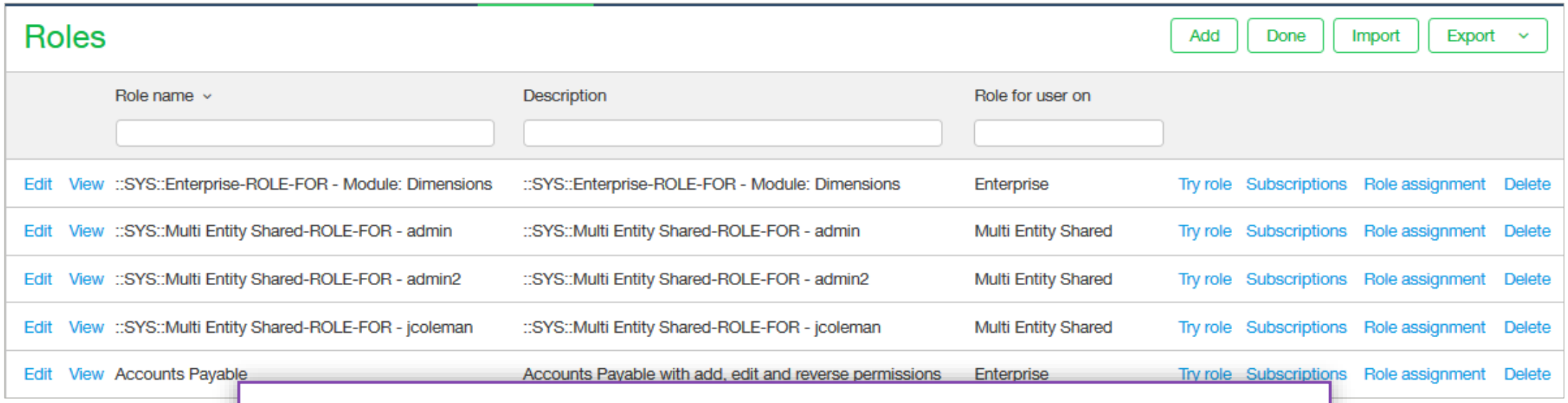
Select Role-based permission type

User-based Permissions to Role-based

Edit, then duplicate roles to rename the role, modify as needed

Roles can be created from scratch or imported

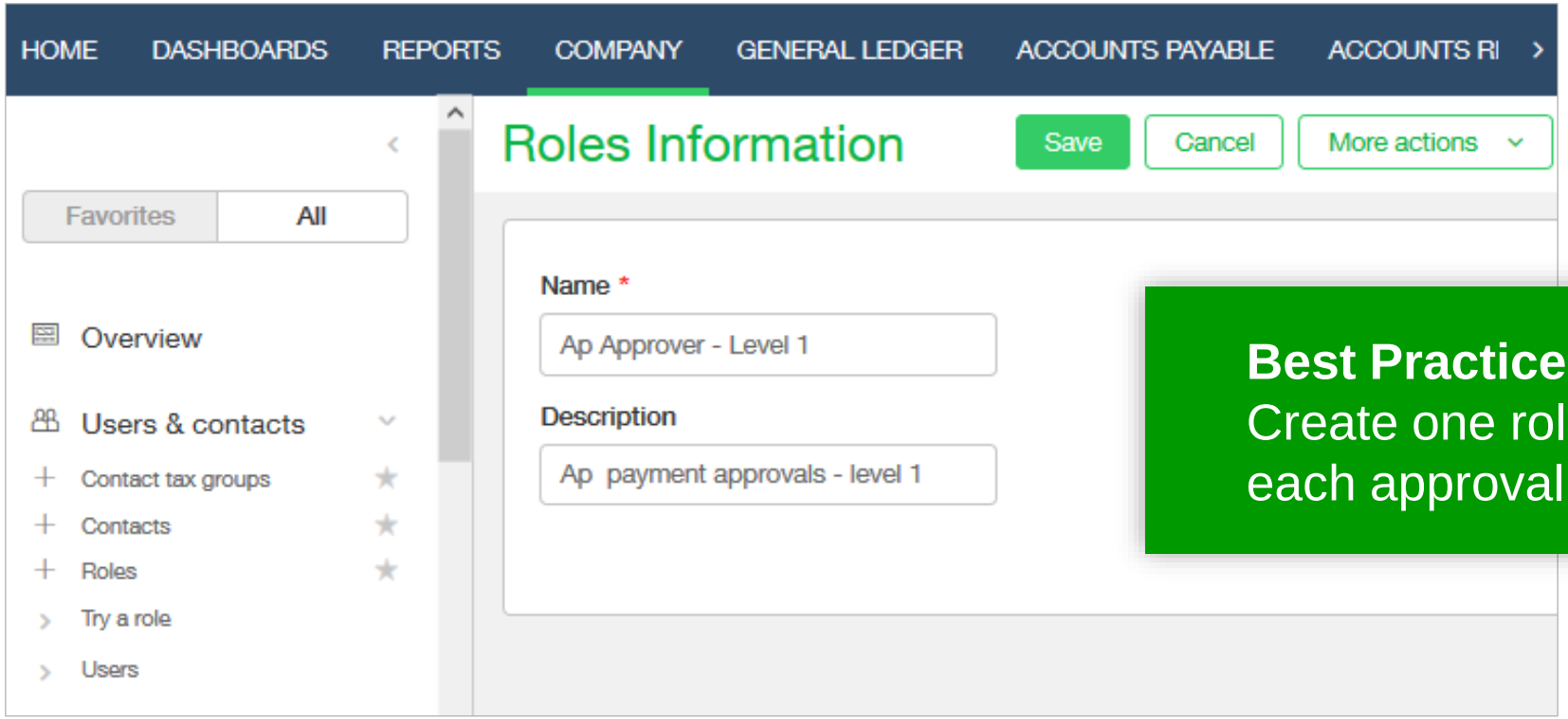
After reassigning roles, unneeded roles can be deleted



Try role applies a business user license

Create a New Role

Company > All > Roles

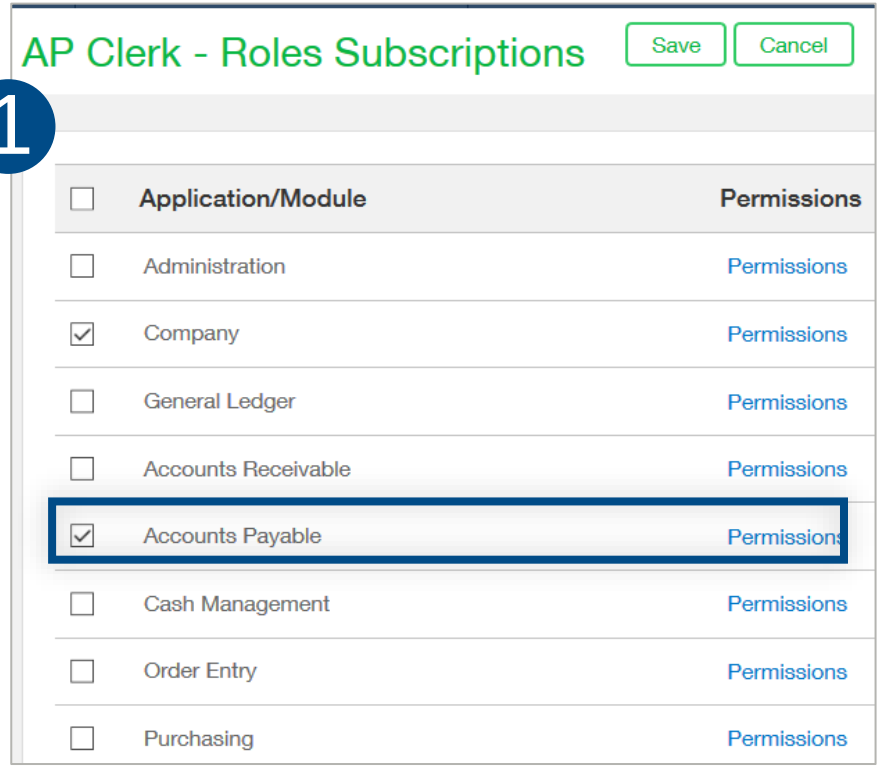


Best Practice:
Create one role for each approval process

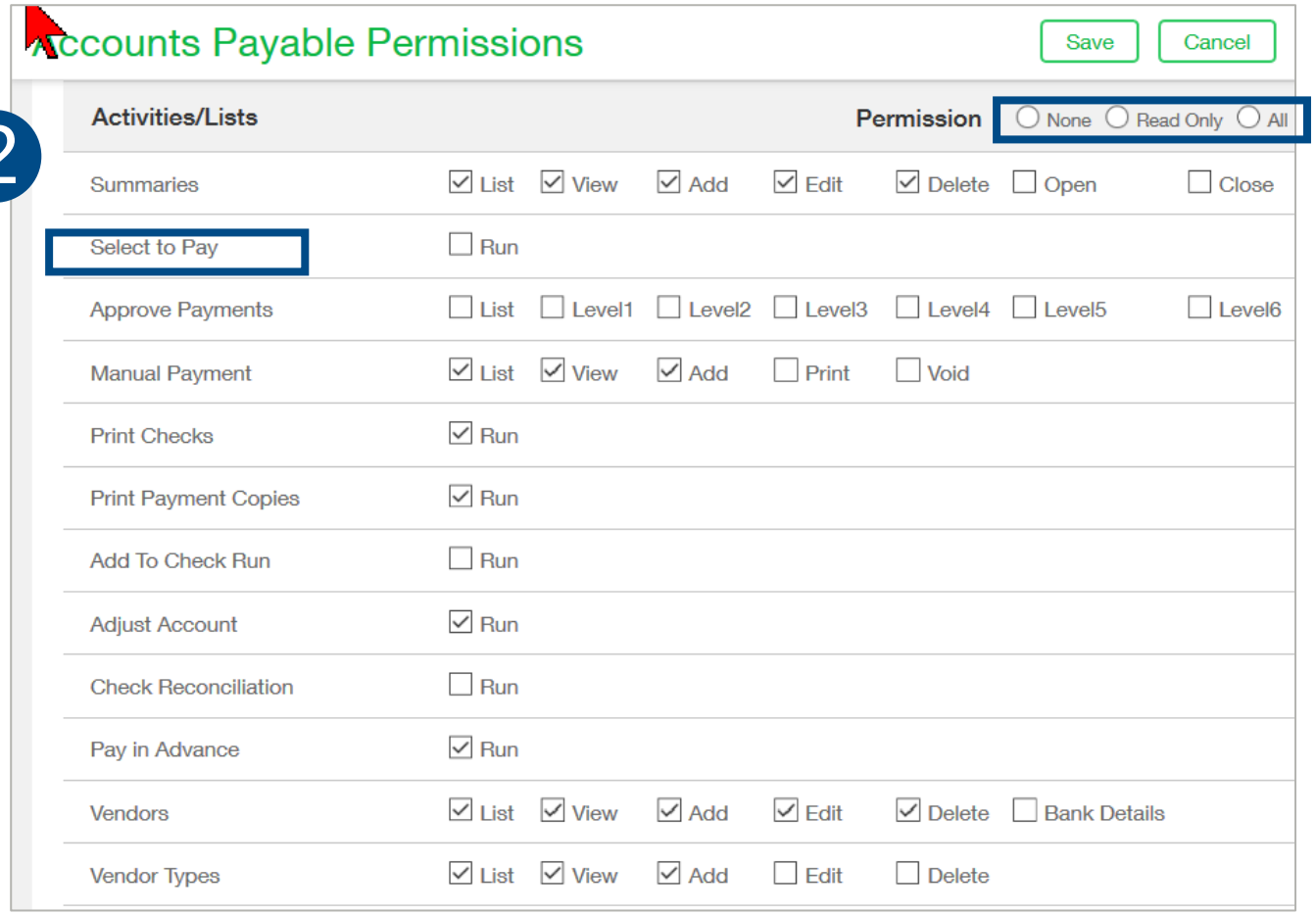
Permissions Review

Permissions are set for each role (or user) application by application

1



2



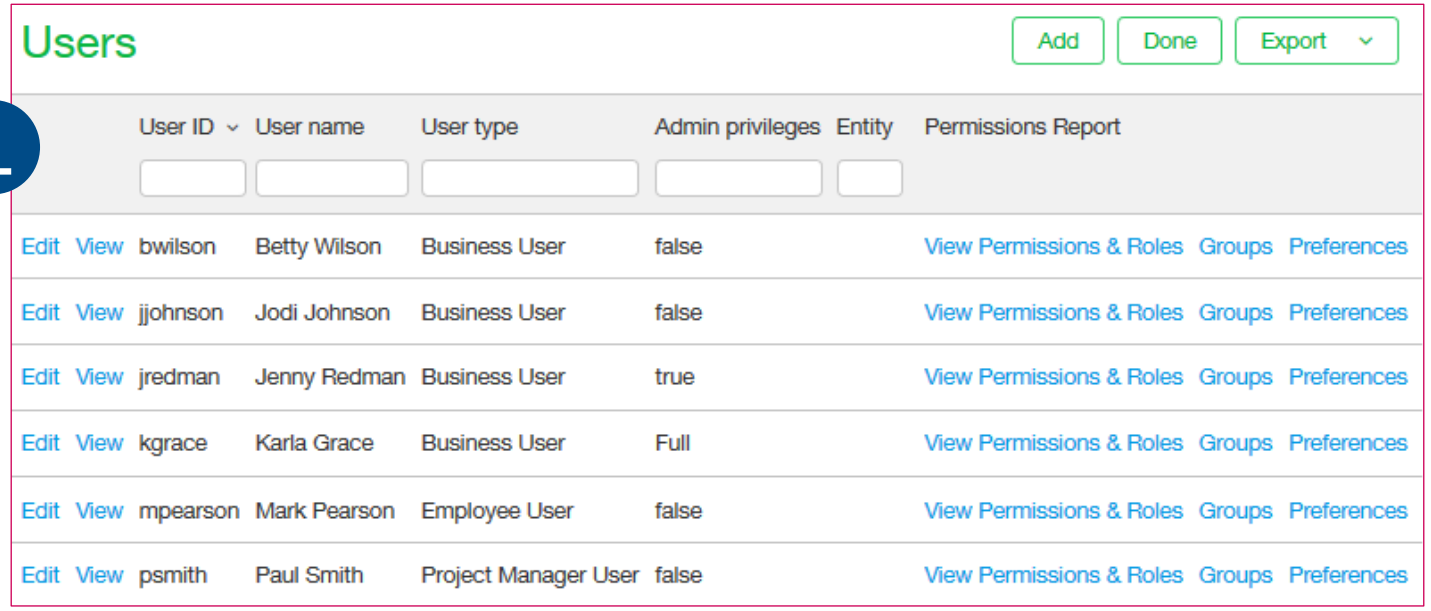
Sage Intacct Permissions

Permission	Description
List	Display the list of data records (e.g. AR > Customers)
View	View the details of a data record
Add	Add a new data record (e.g. add a new customer)
Edit	Edit the details of a data record
Delete	Delete a data record
Reverse	Reverse bills/transactions
Reclassify	Permission to change partially or fully paid bills, invoices and/or adjustments. The changes may or may not have accounting consequences; only in an open period
Run	Permits user access to all screens that are required to perform a function, e.g. Run Reports

Add Roles to a User

Company > All > Users

Roles take effect when the user signs in

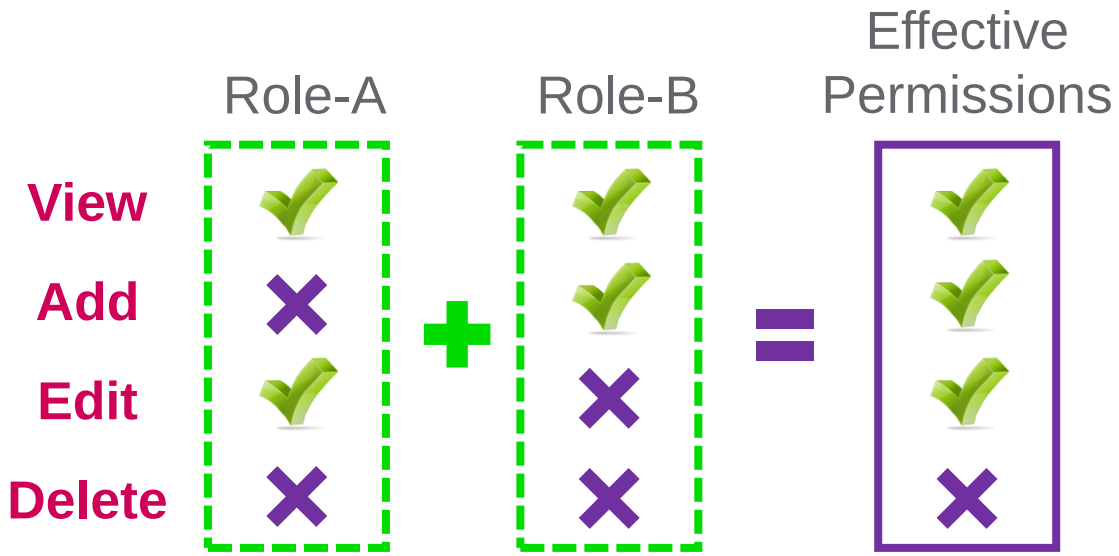


Role Stacking

Permissions for a user with multiple roles

Roles are combined using the “most permission” rule

If one role has granted a permission and another has denied the same permission, the user will have permission



Best Practice for Roles

Create “core” roles that mirror types of users

Create one (1) role if all employees submit both time and expenses; Create two (2) roles if there is a mix (one (1) for time, one (1) for expenses)

One (1) role if there is no separation of duties for AR, AP, expenses; Two (2) or more roles if there is a separation of duties

Use role stacking to layer approval permissions

Create a distinct role for each approval level in each module

Add appropriate roles to users who participate in approvals (NOTE: License types will limit role permissions)

Integration Optimization Considerations

Integration Optimization Considerations

Limit amount of API calls

- No API calls within Loops
- Preload cache files or preload data arrays
- Combine API calls, limit the amount of round trips
- Query 1k records at a time
- Update multiple lines on a given transaction at once
- Consider changing to Per Transaction posting summaries

sage Partner Summit

Thank you