SedonaOffice Release Notes

August 2025

Version 6.2.0.20

Contents

Enhancements/Features

Bill Contact Records Created for All Active Web User Logins for SedonaWeb 2.0

To ensure consistent functionality with SedonaWeb 2.0, this SedonaOffice update adds existing contacts as a bill contact. This looks at all active existing customer contacts with SedonaWeb2.0 access and creates corresponding bill-contact records, so that the new user-access functionality works consistently for new and existing customer portal accounts.

If your customers want to limit access to bills for their contacts, you will need to log into SedonaOffice, edit the contact for your customer, click the Bills and Sites tab, and clear the checkbox for Bills.

The Bill and Sites tab controls what bill to and site information the contact can see when they log into SedonaWeb 2.0. If there are no bill records selected for the contact, the contact will not see any Bill To information like invoices or payment history when logging into SedonaWeb 2.0.

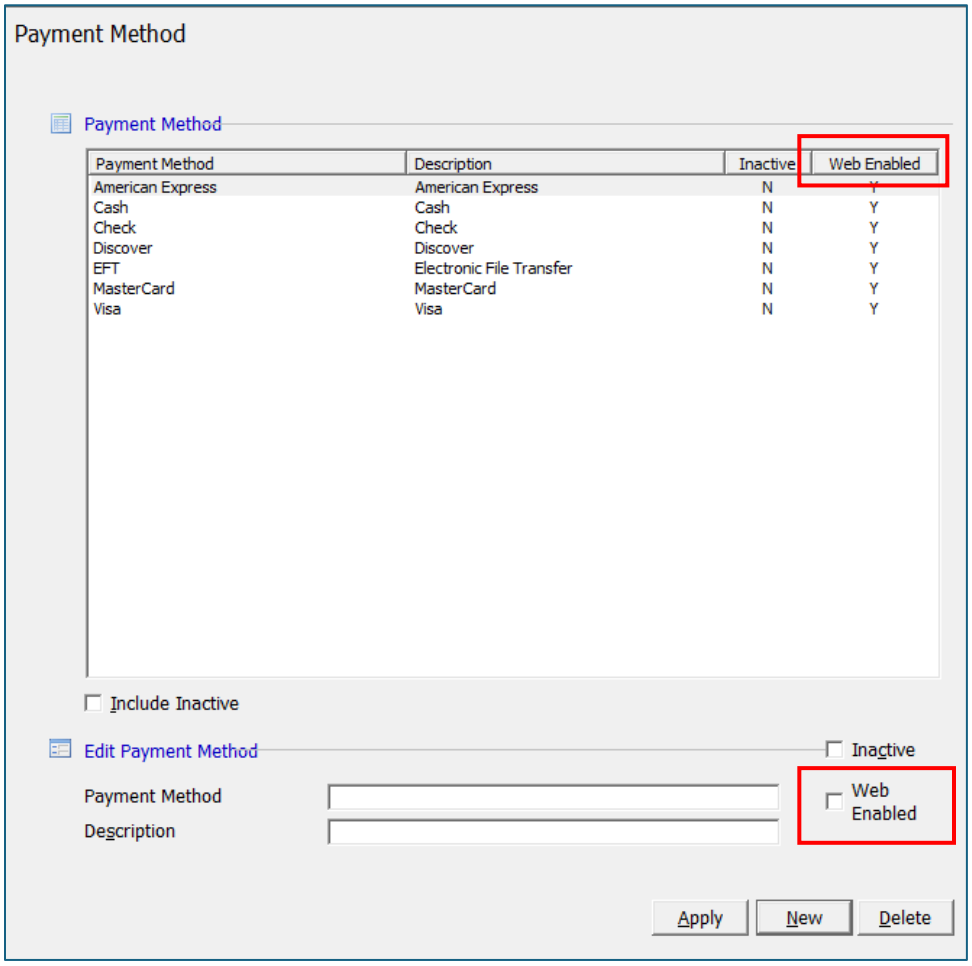
Configurable Payment Options For SedonaWeb [112507]

We added an option in Payment Method settings for configuring SedonaWeb payments. This will work with either SedonaWeb 2.0 or SedonaWeb 1.0:

- SedonaWeb/SedonaAPI 2.0 version 1.51.0 (or later)
- SedonaWeb/SedonaAPI 1.0 version 2.7.83.2 (or later)

This allows SedonaOffice users to select if a payment method should be available for web payments without affecting SedonaOffice payment methods.

Navigate to SedonaSetup > Payment Methods. The changes are the **Web Enabled** checkbox and **Web Enabled** column:

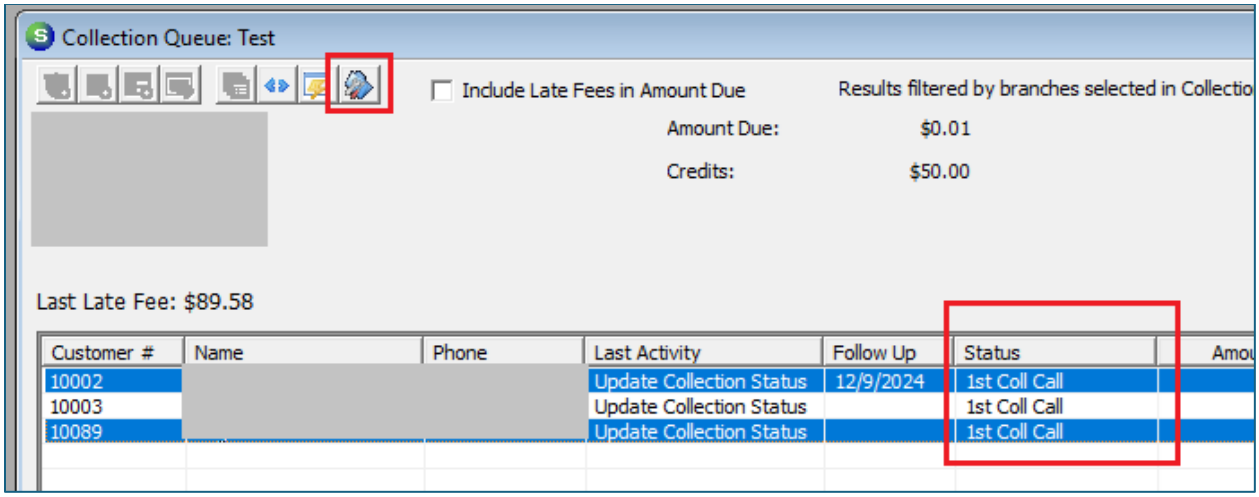


To allow a payment method in SedonaWeb, select the checkbox. The Web Enabled column shows a Y in the row for that payment method. Clearing the checkbox for a payment method, prevents the payment method from appearing in SedonaWeb and removes it as an option for payment for invoices on all payment selection dropdown fields. The Web Enabled column will show an N in the row for that payment method.

Ability to Bulk Update Status Of All Customers Currently Listed In Collection Queue [121280]

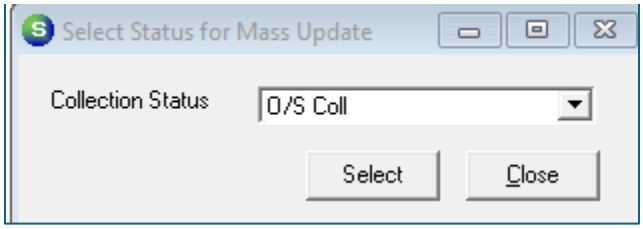
We added a way to update the status of some or all customers selected within a collection queue. Instead of having to update each one individually.

On the Collections Queue form, we added a button, **Mass Update Status**, on the right end of the toolbar:



Select two or more customers from the list by holding down the CTRL key and clicking the desired lines.

Click this **Mass Update Status** button to open a popup to select the status for the mass update:

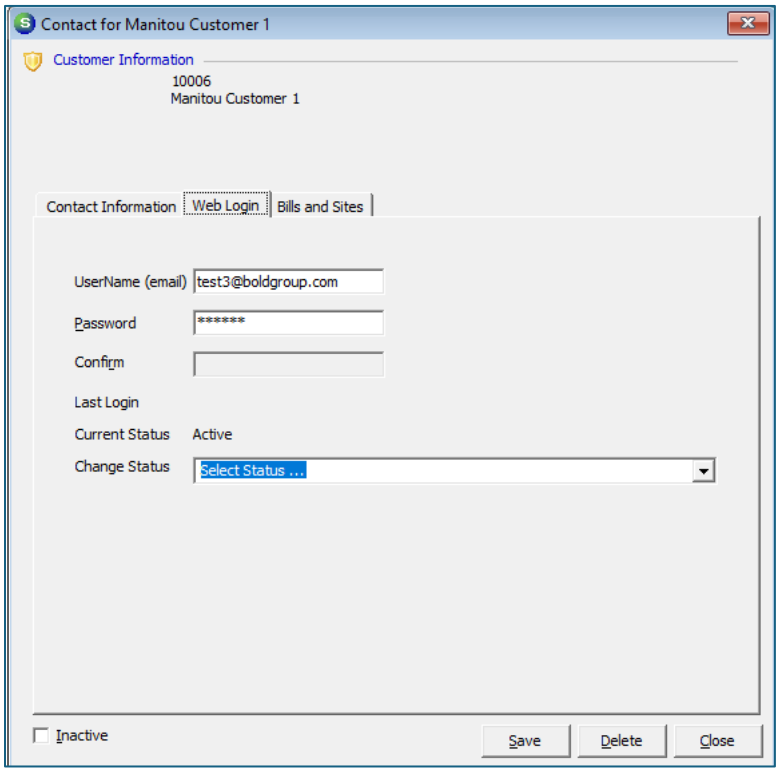


Select a **Collection Status** and click the **Select** button. This closes the popup and updates the status of all selected customers in the collection queue.

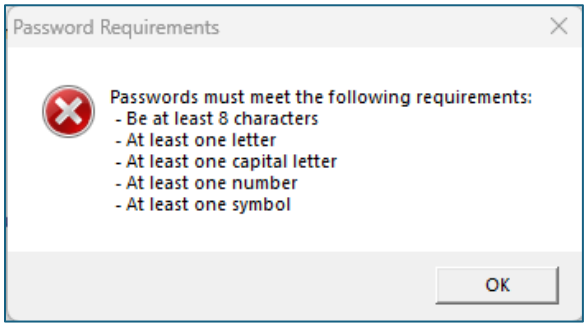
SedonaOffice Customer User Management with SedonaWeb

We added a way for users to manage customer users who are added through SedonaWeb 2.0. SedonaOffice users can add new customer users to the SedonaCloud company associated with their SedonaOffice database; disable customer users from the SedonaCloud company associated with their SedonaOffice database; and set/reset passwords for customer users.

This is on the Contact form on the Web Login tab:



We changed the Password requirements to match the requirements in SedonaWeb 2.0. These are the password requirements:



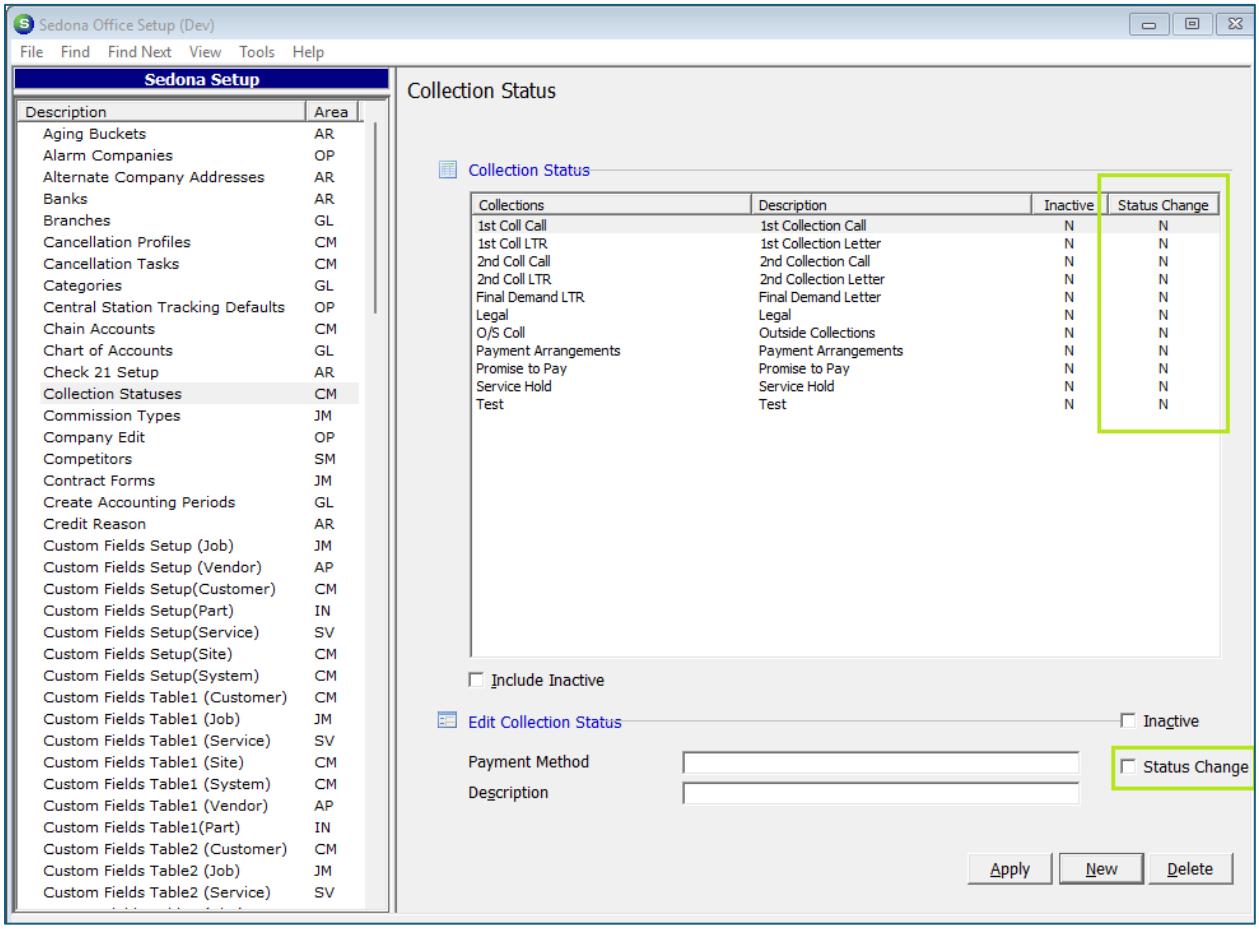
There is a new display only field called **Current Status**. This shows the status of the customer user.

There is a new dropdown field called **Changed Status**. Use this to change the status of the customer user.

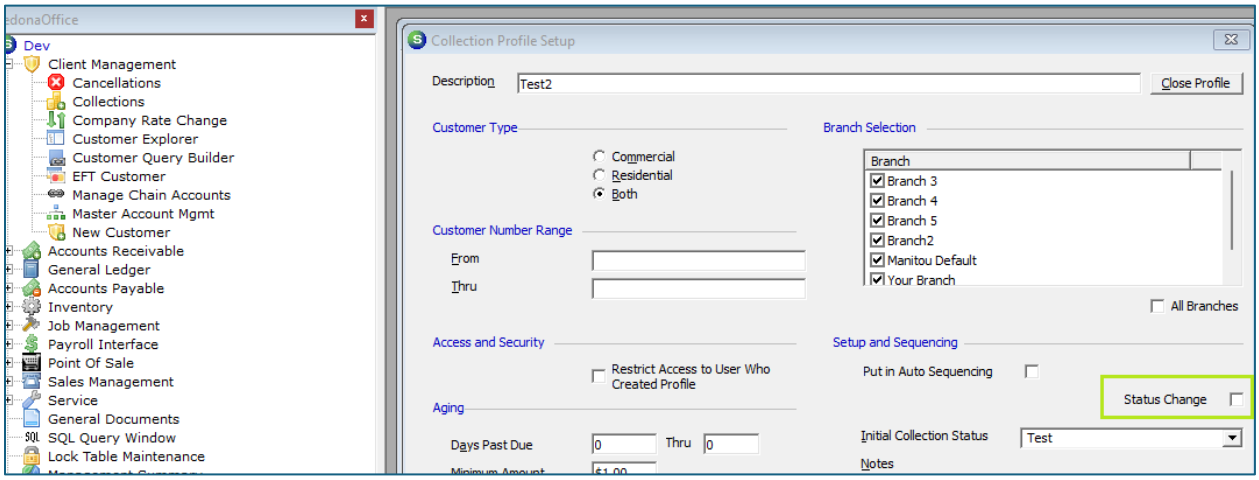
Configurable Automatic Status Changes On Collection Profile [123592]

We added an option for users to either change or not change verbiage to the collection status of accounts when they move through the buckets of the collection queue. By default, an initial status is assigned to accounts as they enter the collection queue. This option allows users to choose that functionality or to change the status as accounts enter a new queue.

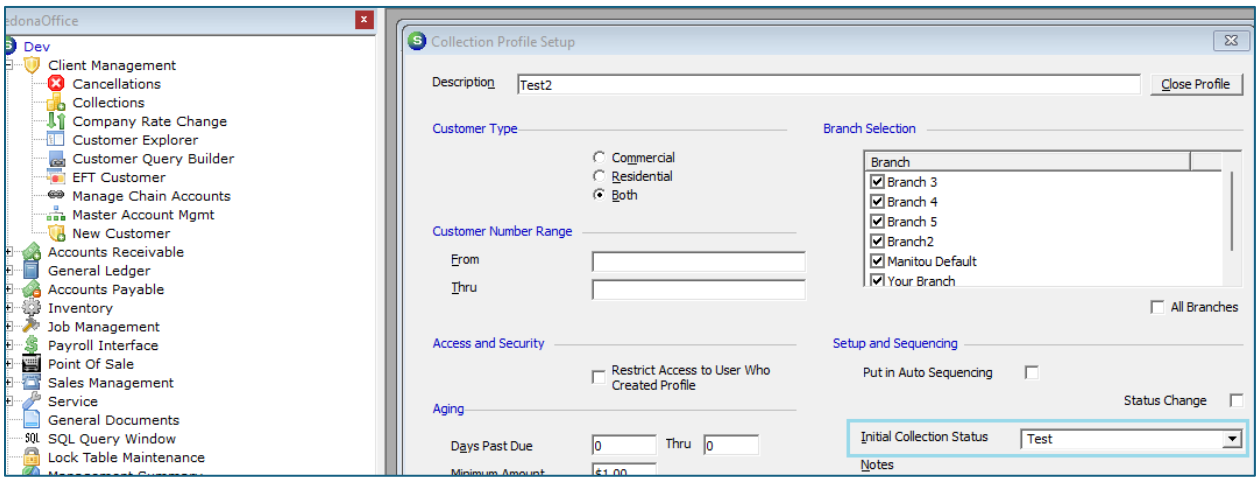
In the Sedona Office Setup > Collection Status, we added a new checkbox and column called **Status Change**:



Under Client Management > Collections > Edit/New Collection Profile Setup, we added a new checkbox called **Status Change**.



When the Status Change checkbox is selected in either of these places, an account's status will take the Initial Collection Status in the Collection Profile Setup when an account enters a collections queue.



The Collection Status for an account will automatically switch when the account goes from one collection queue profile to the next.

Application Corrections

Accounts Payable

Error message trying to edit a vendor [126262]

Users received an error message trying to edit a new vendor. When entering a new vendor, users could input a leading space, and the system allowed saving the vendor. Now starting and ending spaces are automatically removed for Vendor Codes and Vendor Names.

Accounts Receivable

Level 2 CC procurement cards and cycle invoicing

There were two issues:

1. If users created a new Procurement CC on a customer that was set for Advanced Deposit only, it could still be used for Cycle Invoicing.
2. The order_number and sales_tax_amount needed to be included in the in SEFT_Service_Log for these:
   - Service Tickets
   - Cycle Invoices
   - Job Invoices
   - Anywhere that invoices

We made these changes to fix these issues:

- We modified the Payment Form to disable and remove the recurring option if a procurement code is entered and the Invoice checkbox is not selected.
- SedonaOffice will send the tax amount and order number to Forte (SEFT_Service_Log) when processing payments for Invoices, Jobs, Service Tickets, and Recurring transactions using Bank, Credit Card, or Procurement Card.
- When an invoice is partially paid, SedonaOffice will calculate the tax percentage based on the proportion of the total invoice amount that has been paid. This percentage will be sent to Forte for tax purposes.
- For multiple invoices, SedonaOffice will calculate the tax percentage for each invoice individually and sum these percentages.
- The order number will correspond to the first selected invoice.
- Tax amount will not show for forms different than the invoice.

Unhandled Exception: Key not found: 'Selected' Parameter name: Key [129210]

There were occasional instances where users reported receiving an unhandled exception message when trying to add a transaction from a customer card.

This situation was not reproduceable; however, it appears the invoices grid is attempting to resize its columns before the columns exist. In lieu of fixing a known error, we added error handling. Unexpected exceptions are now logged to the desktop Logger.txt file but are not shown to the user. The logger.txt must exist on the desktop before selecting add transaction for the exception to be logged.

We changed the form layout so that if an action cannot be completed, the program will continue without error. For example, if the program attempts to resize the "Selected" column, but it does not exist, nothing is done. There is no message to the user and no error is written to Logger.txt; the program simply continues.

Forte to Sedona card updater issue with expired credit cards and last 4 digits [00127294, 00131954, 129927]

This was fixed with this correction: Locks being created [00137537].

PCI tool erroring on procurement code [00132040]

We resolved an issue in the PCI Compliance Upgrade Tool that caused an error when running it for an upgrade from SedonaOffice 5.x to 6.x.

Non-Recurring autopay issue [00132278, 00133155, 00134069, 00135413, 00136825, 00139528, 00141271]

Users received an error when trying to process the Auto Process Non-recurring Invoices.

We fixed the error that was caused by an incorrect variable assignment.

Procurement code being erroneously added to cards [00133404, 00134962, 00134946, 00135530]

The credit card updater in the EFT service was inadvertently copying the merchant Id to the procurement code field. This has been corrected.

Issues with missing negative payment checks [00133375,00132565, 00134307,00134227,00133889] AND EFT problems [00137611]

We addressed these issues:

1. Voided transactions were not being retrieved from Forte.

2. Processing of Z transactions was failing.
3. Chargeback (C00 settlement) from Forte was not recognized as a reversal. Our ACH transaction was not getting set to “previously funded”.
4. The SEFT_Reconciliation table TransactionStartTime was not adjusted according to the maximum retrieval days. This caused an excessive number of settlements from the past to be retrieved from Forte, causing the run of the SedonaEFTService to take much longer than it should.
5. Procurement code was erroneously added to credit cards.
6. The SedaonaEFTService had a 13 second delay between each payment submission, causing excessive run time.

To address these issues, we made the following changes:

1. Voided transactions were not being retrieved because the API call was timing out. This was most likely because the number of voided transactions was too great to retrieve in one call. We changed this retrieval to use paging so it is done in batches.
2. Setup information was incorrect, causing a generic exception to be thrown and written to the SEFT log. No coding change was necessary to resolve the issue as it was data-related, but we did change the exceptions to contain specific information about what is wrong with the setup data so it will be easier to know exactly what is wrong going forward.
3. We corrected settlement reversal processing to also look for response code C00, which is chargeback.
4. Instead of bumping the start time up by the hourly increment, we changed to set start time to current time minus max retrieval days minus hourly increment.
5. The card updater service was inadvertently copying the merchant ID to the procurement code field when we received an update from Forte.
6. The SQL execution plan was inappropriate for the GL Register balance check. What caused this to happen between runs of the SedonaEFTService is unknown (went from instantly executing the stored procedure to taking approximately 13 seconds), but this situation is typically caused by table statistics being inaccurate. To correct the situation, we modified the Register_List_Balance_Check stored procedure to include table hints so SQL always uses the appropriate indexes.

The following unreported items were also found and addressed:

1. Forte’s “suppress updater” endpoint was being called even when the paymethod had no token. This always resulted in Forte responding with an error because the token was not being supplied in the URI. We changed this so that the call is not made if there is no token or the token is less than 10 characters (which is invalid).

2. A small number of SEFT log entries had only the merchant name and not the merchant Id in the text. When a merchant does not have a name in the setup table, it is not obvious which merchant is being referenced. We added the merchant Id to the messages that had only the name.
3. The SedonaEFTService once-a-day process had SQL timeouts when deleting old log records. We increased the timeout of each applicable query to resolve this.

Receive switch to error when trying to deposit EFT payment batches [00133235]

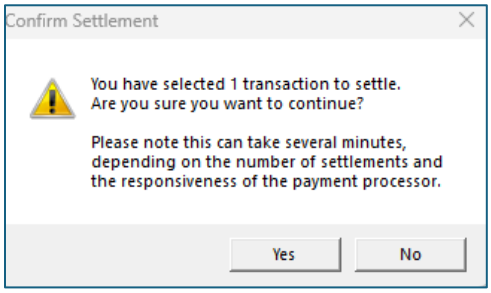
Users received a Switch To Error when trying to deposit eft payment batches, because rejected and refunded transactions had no settled date.

To resolve this, we made these changes:

The SQL query checking for unsettled transactions in the batch was too broad. We modified the query to look only for "approved" transactions.

Resolved the switch to error by running the settlement process in another independent sequence.

Added the "please note..." text to the confirmation message to inform users the settlements can take several minutes so they do not think the client is frozen and stop it before it finishes.



There are 17 orphans transactions from a date that are not correcting themselves [00134718]

This was fixed with this correction: Locks being created [00137537].

3 different scenarios where the system should have generated a negative payment check but failed to do so [00132565]

This was fixed with this correction: Locks being created [00137537].

There are settled EFT transactions with no checks (transactions captured through the API) [00134307, 00133375, 135432]

There were records where the EFT History showed the payment was approved but the payment record did not exist.

Any errors encountered by CaptureTransaction activity coming from SedonaAPI 2.0 endpoints thru the PaymentProcessing dlls will now ensure that AR_ACH records created by the capture transaction logic will be set as 'ORPHANED' status and an error is thrown back to the SedonaAPI 2.0 endpoints.

Locks being created [00137537]

Users were receiving an error message: ERROR: Error processing Response for SedonaCoreApi: status: 400 body: Posting Exceptions occurred.

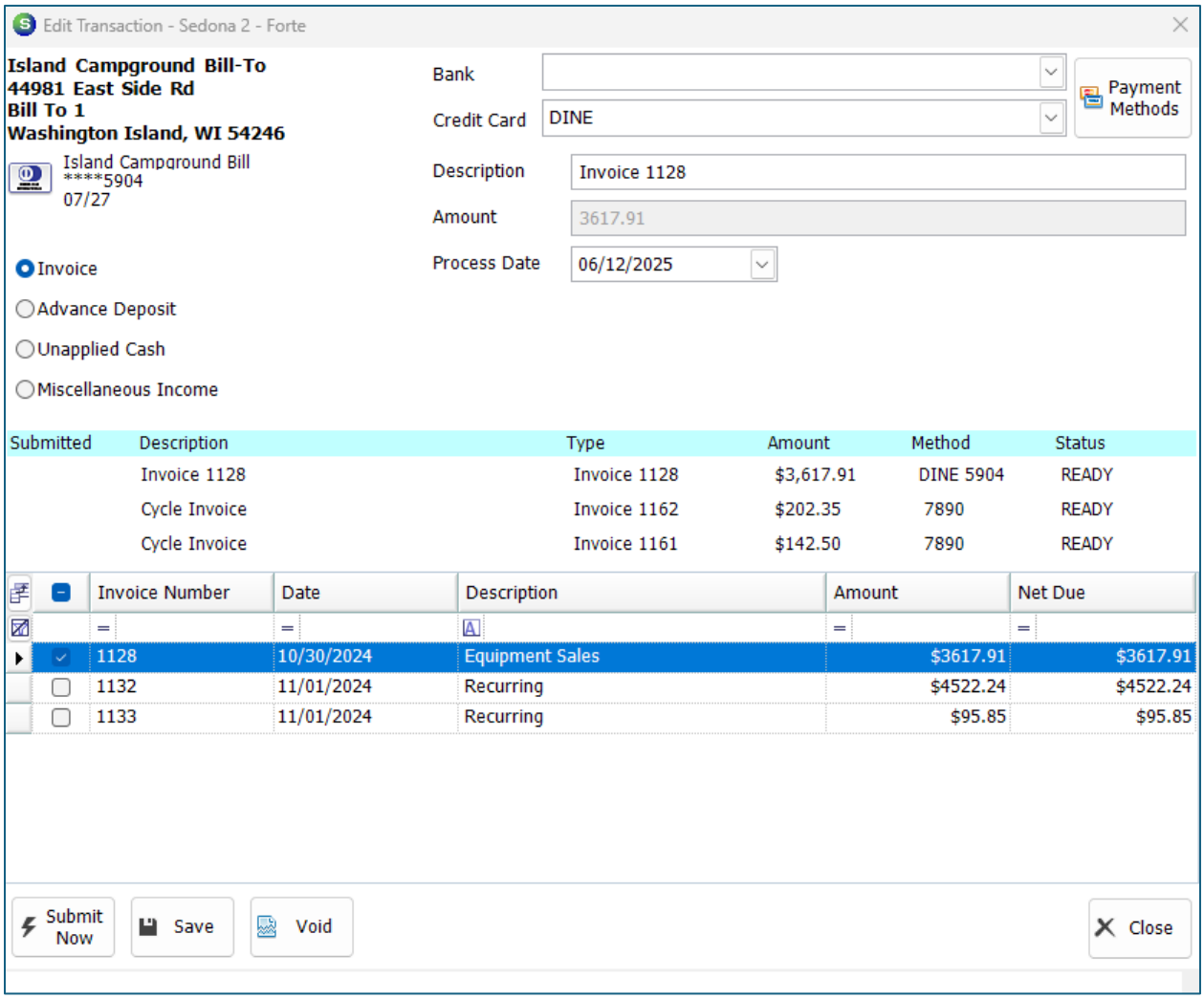
To resolve this, we made the following changes:

1. Changed EFT locks (AR_ACH records plus batches) to take over an existing lock from another user if it has been in place for more than 10 minutes. These locks should normally be in force for only a couple of seconds but could be longer if they are communicating with a payment processor and they are not responding in a timely manner. There are processes that can make up to three API calls to the processor, so a lock could legitimately be in place between 5 and 6 minutes. After 10 minutes, the lock is considered abandoned and can be taken over by another user. This change resolves two issues:
   - Case 00134718 There are 17 orphans transactions from a date that are not correcting themselves
   - Case00137220 Transactions are going to orphaned and never getting updated
2. When the EFT Service submission process is started for an ACH record, ensured the ACH record status is still ready. Literal hours could have passed since the record was first queued, during which time someone else could have changed it (manually submitted, voided, etc.).
3. EFT Service orphaned process:
   - Before processing, ensured the ACH record status is still orphaned. (Same queued scenario as in number 2.)
   - Will not process orphans that are too old:
     - To determine if an orphaned record is too old, the greater of the entered date and hold date is checked and if it less than the current date minus the **max retrieval days** value in the EFT Service config.
     - These are set to void so they no longer get picked up as orphans. This status change is written to the SEFT log as well as the Sedona Event Log. Voiding an orphan clears the EFT pending flag on any associated invoices and detaches any invoices from the transaction.

- This change is in conjunction with number 1. Because old locks will be released (taken over) going forward, old orphaned locked records will get processed after the software update in which these changes are included. To prevent old transactions from getting processed, this change was put in place. Going forward, under normal circumstances, there should never be any orphans that are too old, but we will properly handle them if they ever exist.

4. When a Forte transaction with our ACH Id reference is not found, we will also search Forte transactions by their transaction Id. This is especially applicable to situations where the payment transaction is authorized outside SedonaOffice and then capturing the authorized transaction via SedonaWeb 2.0 API fails before posting is complete. In this case, our ACH transaction will be orphaned, but Forte will not have our ACH Id reference. Therefore, we would otherwise not be able to find the Forte transaction applicable to our ACH record. This change resolves the issue of duplicate payment submissions that was uncovered when fixing another issue.
5. A Forte settlement transaction will no longer be written to the SEFT log if it has already been processed. The log was cluttered with inconsequential information.
6. A Forte funding settlement transaction will no longer be written to the SEFT log if it has already been processed. The log was cluttered with inconsequential information.
7. Ensured each AR_ACH record is properly locked/unlocked for exclusive use during payment submission, orphan processing, settlement processing, funding processing, and void processing.
8. Updated the Register_Balance_Check stored procedure to call the optimized Register_List_Balance_Check stored procedure. Both had the same functionality. One checked a single register number while the other checked a list of register numbers. We used the Register_List_Balance_Check stored procedure to check a list of one, thereby having the functionality in only one place.
9. Updated the capture transaction functionality used by SedonaWeb 2.0 API:
   - Implemented proper ACH record locking.
   - Ensured an ACH record is marked orphaned until posting is complete.
   - Ensured the Forte transaction Id is immediately stored in the ACH record's trace number so it can be used to find the Forte transaction without using our ACH Id number.
10. Added SEFT logging to the card updater service to show what we got from Forte. (This will help prove issues such as the 1930 CC expiration year.)
11. The EFT transaction form was obtaining the ACH record lock as soon as the form was opened. This allowed the lock to be in place for an indefinite amount of time and would mess up the new 10-minute rule. Changed this to not obtain a lock until an action is taken (submit, save, void), at which time the current ACH record data is retrieved from the database. If the current transaction

status does not equal the original status from when the form was loaded (in case somebody else made a change), the change is not allowed.



12. Changed the EFT transaction form to not exclude credit cards that are expired from the drop-down list. It is possible that the expiration has been updated by the card provider but has not been updated in SedonaOffice yet. It is up to the provider to determine if the paymethod is valid when submitted This resolves *cases 00127294, 00131954, 129927 Forte to Sedona card updater issue with expired credit cards and last 4 digits*.
13. Changed the EFT transaction form to show a message (with a countdown timer) to the user when waiting more than 3 seconds for a response from the payment processor (Forte).
    - The countdown time depends on the operation being performed. For example, a payment submission can make up to three API calls, each of which can be active for up to 100 seconds

(after which it will time out), plus a total of 4 seconds of delay between the calls. A few more seconds of buffer time are added to the total possible API call time.
    - This functionality is used on a payment submission and when voiding an approved transaction, as that goes out to the processor to void it there.
    - The countdown timer should never reach zero due to the buffer time added, the API call(s) should time out before zero is reached but is there as a failsafe. If zero is reached, the payment processor call will terminate and the submission/void will fail in the same manner as if the API call itself timed out.
    - The countdown timer was added so users do not think the client is stuck and close it, which will cause data cleanup to not occur, including causing locks to remain.
14. Changed the EFT transaction and EFT processing forms to allow orphaned transactions to be voided.
15. Changed the EFT processing form in the same manner as the EFT transaction form during submission/void processes (ACH record locks, countdown form).
16. EFT Processing form > Approved tab > Settle:
    - This was only retrieving the last settlement transaction from Forte. This was incorrect as all unprocessed settlements must be applied to our ACH record, especially because we are not assured that Forte sends us the settlements in the appropriate chronological order. Changed this to retrieve and process all settlements for the given transaction reference.
    - Added a please wait message box with a spinner to show users that the client is still active so they do not think it is stuck. This does not show exactly what it going on (which ACH Id is being processed, etc.) because of the way the payment processing projects are configured.
17. Added a new EFT transaction showing ACH Id 0 in the customer event log:

Time Stamp	User Code	Type	Description
6/16/2025 11:16:14 …	bold	ADD	New EFT Transaction 14.91. (ACH Id 0)

18. During the settlement process, when switching from previously funded to settled, the dispute records are retrieved from Forte. The dispute record status text is used to update the deposit check memo field to show what happened with the chargeback. The first (earliest) dispute record's status was always being used for this. If there are multiple back and forth previously funded/settled changes, it makes more sense to use the most recent dispute status. Changed to do so.
19. During the settlement process, the response code was required to be blank for the Forte settlement record to be considered a settled record. This is always the case for the original settled record. However, a re-settle record (settled > previously funded > settled) had a response code of

U00, which is not listed as a valid code according to Forte's documentation, and U codes are supposed to indicate rejected:

Transaction Response Codes

When a transaction is submitted for processing, Forte immediately returns one of the following responses. Transactions that are accepted for processing return the **A** response codes and transactions that are pending for processing return the P response codes. (Note: P codes are currently applicable only for PayPal transactions). Transactions that are rejected for processing return the **U** response codes. Transactions that have formatting errors in the message return the **F** response codes, and transactions that run into exceptions return the **E** response codes.

Code	Description	Comments	Test Parameters
		Declined	
N01	NO INFO	The Routing and account numbers are found in the database, but no names are associated with these.	
U02		Customer account is in Forte's "known bad" account list (EFT only)	Send echeck sale transaction with the following data: 1. routing_number = 021000021 2. account_number = 987654321
	TRN NOT APPROVED	Routing number passes checksum test but not valid for ACH	Send echeck sale transaction with the following data: 1. routing_number = 064000101 2. account_number = Any account number
U03	DAILY TRANS LIMIT	Merchant daily limit exceeded (EFT only)	Not available
U04	MONTHLY TRANS LIMIT	Merchant monthly limit exceeded (EFT only)	Not available
U05	AVS FAILURE ZIPCODE	AVS state/zip code check failed	Send a regionand postal_code that do not match
U06	AVS FAILURE AREACODE	AVS state/area code check failed	Send a regionand postal_code that do not match
U07	AVS FAILURE EMAIL	AVS anonymous email check failed	Send an email from a hotmail.com email address.

- It is not known why Forte would send that code on what should be an approved, settled record. We will ignore the response code when determining if a settlement is settled and go by the deposit/withdrawal settlement type.

- We made the same change when determining if the settlement is a reversal; we will ignore the response code and go by the deposit/withdrawal settlement type.
- The EFT service no longer attempts to process orphans older than the **max retrieval days** value in the EFT Service config. The EFT service sets them to void, so they no longer get picked up as orphans. This status change is written to the SEFT log as well as the Sedona Event Log.

Transactions are going to orphaned and never getting updated [00137220]

Some transactions were going to orphaned and never getting updated.

This was due to locks in the SS_LockTable. We resolved this issue when correcting the issue in this: Locks being created [00137537].

Voided payment showing as approved in Sedona [00134227]

This was fixed with this correction: Locks being created [00137537].

Client Management

Old master account name still showing after change [00071420, 82578, 92678, 94420, 120657, 122183, 128514]

We fixed a situation where master accounts were not being updated when customer information was changed.

A/R - sub-account billing box not being auto check marked [113752]

We fixed an issue where the Bill to Primary Master for sub accounts did not show as selected.

Updating 'next' sub number on master account [00130641]

When changing the Number in the Account Index Field on a Master Account, that field did not save the Number entered.

We fixed this so users can enter and save the number entered in the Account Index field.

Integrations

ADI integration has increased the AP invoice field length from 8 to 10 [00142719]

For ADI integrations, the Invoice number field length was changed from 8 characters to 10 characters.

Setup

Service ticket message in Sedona Setup does not allow enough characters

Service Ticket Message in Sedona Setup did not allow enough characters. The SV_Setup.ticket_message field in the database is nvarchar(128), but users could not enter that many characters on the screen or on the printed ticket. The screen allowed a maximum of 90 characters.

We expanded the input field on the screen from 90 to 128 characters.

Stored Procedures

New Stored Procedures

None for this release

Updated Stored Procedures

- ACH_Add
- ACH_Cycle
- ACH_Invoice_Add
- ACH_Invoice_Del
- ACH_Invoice_Upd
- ACH_Refresh
- ACH_Upd
- AR_ACH
- Collection_Profile_ADD
- Collection_Profile_GetOne
- Collection_Profile_UPD
- Collection_Queue_ADD
- Collection_Status_ADD
- Collection_Status_UPD
- Customer_CC_Add
- Customer_UPD
- Register_Balance_Check
- Register_List_Balance_Check
- SEFT_ACH_Add
- SEFT_Get_ACHTransactionByID
- SEFT_Get_ACHTransactionByTraceNumber

- SEFT_Get_ACHTransactions
- SEFT_LockTable_Acquire
- SEFT_LockTable_Release (removed)
- Service_Ticket_Part_GET2
- WS_Account_ADD
- WS_Account_ADD2
- WS_Account_DEL
- WS_Account_Register
- WS_Account_UPD

Database Tables

New Database Tables

- WS_Account_DeleteSC

Updated Database Tables

- AR_Collection_Status
- AR_Payment_Method
- WS_Account
- WS_CustomerList

Special Upgrade/Installation Instructions

If upgrading from a SedonaOffice version prior to 6.0, the following related updates are also required:

Legacy SedonaWeb 1.0 — Be aware that if your company uses Legacy SedonaWeb 1.0 (version 2.7.80 or earlier) with SedonaOffice version 6.2.0.8 or earlier, we recommend that you transition to using SedonaWeb 2.0. (Note: Legacy SedonaWeb 1.0 version 2.7.81 is compatible with SedonaOffice 6.2.0.9 or later.)

SedonaWeb/SedonaAPI 2.0 Setup — If your company uses the SedonaWeb/SedonaAPI 2.0 in any manner (Sales Automation, Time & Attendance, eForms, or the SedonaAPI for integrations such as the Manitou integration), IT will update your SedonaWeb/SedonaAPI version at the same time as your SedonaOffice version. This is to ensure compatibility with the Sales Automation module.

Performing Update — Once you have reviewed all the above information, and followed all preparation steps, contact SedonaOffice support. We will note on your account that you have received the Release Notes and are ready for update. SedonaOffice IT will then contact you to schedule your update.

To Use TLS 1.2 — Consider the following:

- All computers running SedonaOffice client must be on Windows 10 with the October 20, 2020 build, version 17763.1554 or later; the server must be on Windows Server 2019 or later.
- TLS 1.2 must be the only TLS version enabled in the Registry. TLS 1.0 and TLS 1.1 must be disabled. Verify that they are disabled and that TLS 1.2 is enabled.
- The SQL Server must be set to force encryption.

Supported Environments

Minimum System Requirements

- Server is on Microsoft .NET 4.6.1
- If used, SedonaWeb/SedonaAPI 2.0 version 1.51.0 (or higher)