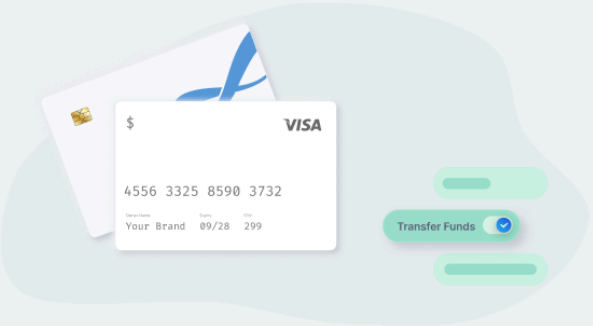
cross river | Docs

Concepts

1. Concepts intro

Explore our concepts

Cross River's platform capabilities can power and scale your fintech applications.



Learn how Cross River's platform capabilities can support and grow your fintech applications. Our concept overviews break down each product, showing you the main features and how to integrate them. Whether you're starting from scratch or improving what you have, these resources help you understand how everything works and make the most of our solutions.

Accounts

Robust set of API calls to create, update, and manage your loans.

Checking
For everyday transactions

Savings
Various deposit accounts

CDs
Lock funds for higher interest

FBO
For the benefit of accounts

Cards

Scale debit, credit, and pre-paid card programs with processor-agnostic issuing platforms.

Partner-managed
Using your preferred processor

Integrated issuing
Full-service integrated issuing

Payments

Streamlined payment solutions encompassing both local and international transfers.

Instant payments
Clear and settle immediately

Card payments
Transfer funds to/from cards

International payments
Send funds across the globe

ACH
Originate standard and same day

Wires
Transfer through Fedwire

XML batch payments
ISO20022 payment instructions

Checks
Check writing and deposits

Lending

Robust set of API calls to create, update, and manage your loans.

Marketplace lending
Loan solutions

Digital banking

Manage your accounts easily and securely with our digital banking platform.

Digital banking
Digital banking platform

2. Accounts

Accounts APIs

Cross River delivers a number of different options and configurations for bank accounts you use or you offer your own customers. Bank accounts are a way for individuals and business to store, receive and use funds in various ways. For some purposes you or your customers must have a bank account with Cross River, for example, to make or receive **payments**.

What Cross River offers

Cross River offers several account products:

- Our **Deposit Account** products include:
  - **Checking**: A transactional account. It is designed for individuals to deposit money into it and take money out of it frequently. People usually use checking accounts to keep their money available for paying bills and withdrawing money for regular use. Money in a checking account may earn interest.
  - **Savings**: Used to set aside money for the future. People usually use savings accounts to deposit money they do not expect to need or use on a regular basis.
  - **CDs**: Another type of account consumers can use to set aside money for the future. Unlike a savings account, you cannot withdraw funds from a CD whenever you like. Typically, you must keep money in a CD for a certain period of time or you will likely have to pay a penalty or lose some or all of the interest you earned. The required period of time could be three months to five years or more. In addition, there may be a required minimum deposit. CDs typically offer a higher rate of interest than regular savings accounts.
  - Our **deposit sweeps** feature gives your customers a higher insurance limit. deposit sweeps can be used with both checking and savings accounts.
- **FBO**: An FBO or "For benefit of" account is a special type of bank account that you, as our partner, can set up for the benefit of others. These accounts can be for one entity or for many people combined.

- **Subledgers** (virtual accounts): Cross River gives you the ability to create subledgers instantly and on demand. A subledger can be linked to any existing deposit account, also known as the master account. Subledgers (sub-accounts) let you keep track of daily transactions in your account and to ensure that funds exist where expected and allow for easy end-of-day reconciliation. They are used to receive funds from and send funds to customers without having to expose your master account number.

The product type determines the behaviors and configurations of these accounts.

Find out more

- **Accounts in COS Explorer**

Tutorials

- **Open an account**
- **Withdraw CD funds early**
- **Open a subledger**

2.1. Checking and savings

Accounts APIs

Cross River provides several different types of account types, including savings accounts, Certificates of Deposit (CDs or time deposit accounts) and demand deposit accounts. Accounts can be either interest-bearing or non-interest bearing.

Contact your relationship manager to define what types of accounts are needed for your offerings.

Checking accounts

Check management APIs

By default, all Cross River partners have checks blocked (turned off) for their products. This means a customer can order checks using a third party and issue a check, but that check will automatically be rejected by Cross River when presented by the Federal Reserve for payment. You can request that Cross River enable checks per product or even per specific account. If you want to offer checking to your customers on an individual basis, you can keep checks blocked for the product and have Cross River enable you to have each customer opt in individually.

Learn more about **checks** in our payment products section.

2.2. Certificate of deposit (CD)

Certificates of deposit (CDs), or time deposits, are a type of deposit account. CDs are sold in specific amounts and with specific maturity dates. If the customer decides to withdraw the funds before the applicable maturity date, they will typically be required to pay a penalty.

A *penalty month* is a configuration that defines the maximum number of months that a penalty applies. For instance, configuring penalty months as **12** would mean that if the money is withdrawn (account is closed) before 12 months have passed, (x) fee days would be withdrawn from the account as a penalty.

Fee days define how many days of simple interest to assess as a penalty. COS allows a maximum of 2 penalty tiers for CDs.

Create a CD account with the **open a deposit account** endpoint. You must have the proper product ID and a customer ID for the account holder.

Other CD endpoints available include:

- **Withdraw funds early from a CD**
- **Get activity on specific CD**
- **Find out the penalty for early withdrawal from a CD**

2.3. For benefit of (FBO) accounts

An FBO account is a type of bank account that can be set up for the benefit of others. These accounts can be for one person or entity or for many.

FBO types

The structure of an FBO account to support your needs depends on your specific use case. As you are onboarding, Cross River will work with you to determine the best structure to support you.

Here's an example of how an FBO account is often structured to support a fintech use case:

1. **The fintech opens a single bank account** with a partner bank. The account is titled something like: *Fintech Inc. FBO [Its Users].* This shows the bank that the account funds don't belong to the fintech itself—they belong to its users.
2. **The fintech keeps a ledger**—a digital record—that tracks how much of that pooled money belongs to each user. The bank doesn't see or manage this breakdown; it only sees the total balance in the FBO account.
3. **The account is not FDIC-insured per individual user by default**, but it *can* be set up to do so if the fintech and the bank meet certain pass-through requirements, like proper user identification and record-keeping. If done correctly, each user's funds in the FBO are protected just like a personal checking account would be.

Advantages and use cases

FBOs are widely used in fintech because they:

- Avoid the need to individually open and manage thousands or millions of individual bank accounts
- Still comply with financial regulations, like storing customer funds separately from company funds

Common use cases include:

- Digital wallets (for example, Venmo or Cash App-like apps)
- Investment platforms
- Marketplaces holding seller/buyer funds
- Earned wage access or early pay apps

2.3.1. Subledgers

Subledgers APIs

Cross River gives you the ability to **open a subledger** instantly and on-demand. A subledger can be linked to any existing deposit account, also known as the master account.

> In subledger API endpoints, the term *subaccounts* is used instead of subledgers. For instance, `POST /v1/dda/subaccounts`.

In the Cross River system, a subledger is a first class citizen, meaning it supports all the operations available to accounts. All transactions first go through a subledger, followed by the master account. Each master account has what is called an implicit subledger, which is automatically created and maintained. The implicit subledger accounts for the portion of the balance which is directly attributed to the master account's activity. The Cross River system does this by giving it the same account number as the master.

With the use of an implicit subledger, master account subledgers always sum to the balance of the master account. It is virtually impossible for your master account and subledger to be out of balance.

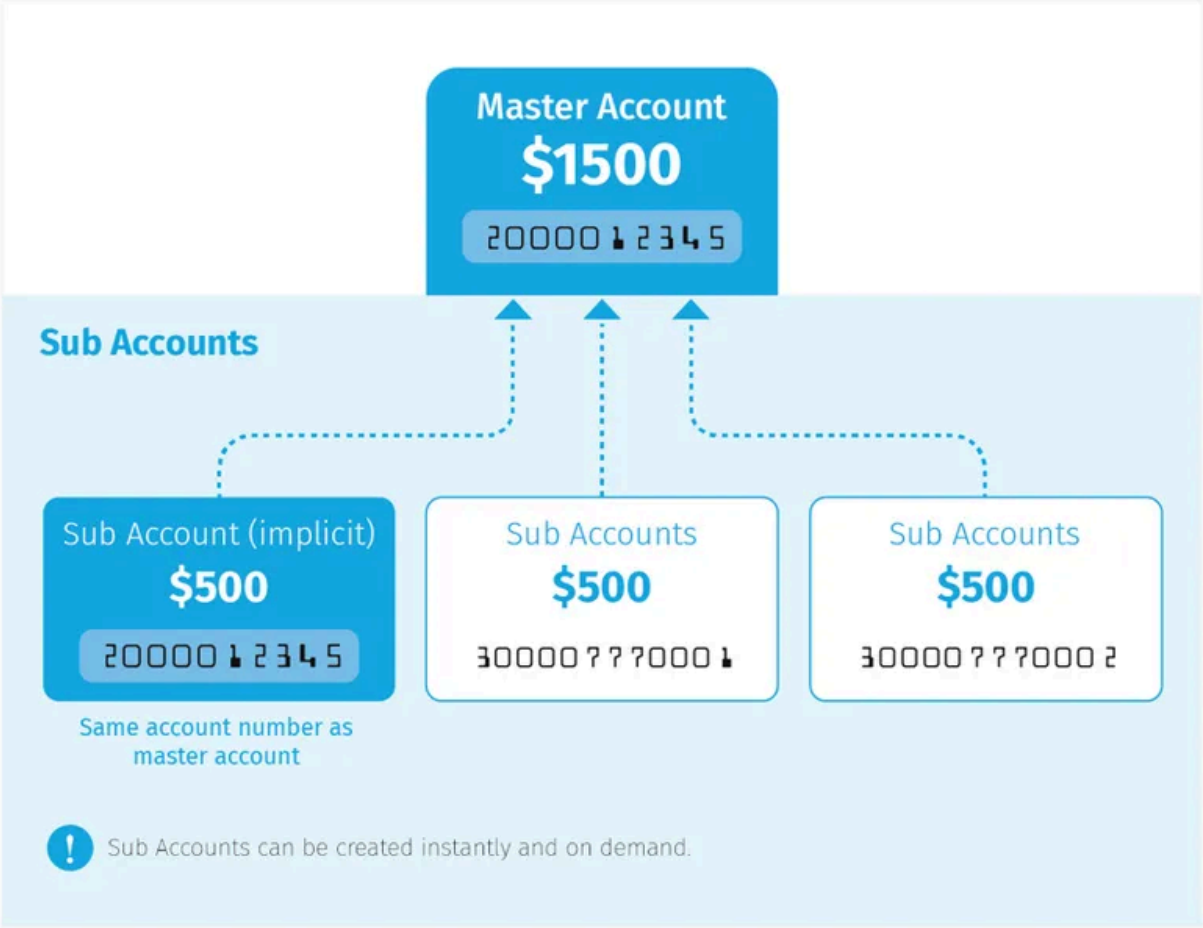
The master account implicit balance can go negative. This would occur in situations where more funds are attributed to other subledgers than the master account directly. It is the responsibility of the partner to maintain and reconcile the subledger system.

> The subledger and master account balances are updated atomically and never out of sync. Funds posted to a subledger are immediately available

Subledger structure

The following example explains the relationship between subledgers and a master account.

Sample Sub Ledger



1. A checking account is opened with an initial deposit of $500. An implicit subledger with the same number is created and is tracking the balance of the master.

Master Account #	Subledger Account #	Balance
2000012345	--	$500
	2000012345 (*Implicit*)	$500

2. Next, two subledgers are created at the request of the partner:

Master Account #	Subledger Account #	Balance
2000012345	--	$500
	2000012345 (*Implicit*)	$500
	300007770001	$0
	300007770002	$0

3 A wire for $500 is sent to each of the subledgers:

Master Account #	Subledger Account #	Balance
2000012345	--	$1500
	2000012345 (*Implicit*)	$500
	300007770001	$500
	300007770002	$500

4 $1,000 is pulled from the master account directly:

Master Account #	Subledger Account #	Balance
2000012345	--	$500
	2000012345 (*Implicit*)	($500)
	300007770001	$500
	300007770002	$500

The implicit subledger is now negative but the sum of the subledger still correctly reflects the master account balance.

> Transfer subledger funds to the master account before spending the money. The implicit balance should match the master balance.

Subledger types

Passthrough subledgers

With passthrough subledger configuration, the master account balance is the true representation of value in the account when determining funds availability for transactions. Thus only the master account balance will be used in transaction decisions. The master account balance is not dependent on subledgers, and the master account balance can move above or below the aggregate value reflected in the subledger balances. Thus the sub ledger balance is ignored.

An individual subledger balance in this configuration can go negative, as can the implicit subledger balance. However, the master account must have a sufficient balance for a withdrawal to post.

Usually, with a passthrough configuration the partner maintains a ledger with end-user (customer) balances on its own systems. The partner can also maintain a customer ledger at Cross River with the passthrough configuration. However, that requires that the partner actively maintain a true balance by transferring funds to and from the master and subledgers as needed (for example, a withdrawal from the master account requires a corresponding transfer from a subledger to the master account).

For this reason, passthrough subledgers aren't an optimal way to maintain a balance of all their customers' dollars when compared to the direct subledger configuration.

Withdrawing funds

With passthrough subledgers, partners can withdraw funds from the master account or directly from a subledger.

Withdrawing funds from master account

When funds are withdrawn from the master account, they are debited from the implicit subledger (which represents the true balance), as well as from the master account. This is because in the passthrough configuration the aggregate value in the non-implicit subledgers does not equal the total balance in the master account.

Partners can withdraw funds from the master account without first transferring funds from a subledger to the master account. In this case, the subledger will represent a balance which is not a true representation of available funds.

Withdrawing funds from subledger account

If a partner wants to withdraw funds directly from the subledger, there is a corresponding debit to the implicit subledger. If the master account is designated in the API as the account to withdraw from, funds will be debited from the master account and the implicit subledger. If a subledger is used in your transaction or payment request, the Cross River system will transact on the master by that amount.

Depositing funds

In passthrough, when funds are deposited they are credited to the designated subledger, the implicit subledger, and the master ledger.

Direct subledgers

Under a direct ledger configuration the aggregate subledger balance will be used in transaction decisions. The implicit subledger cannot go negative, and neither can any of the individual subledger balances.

With direct subledgers, the aggregate sub ledger balance must equal the value in the master account. If the partner wishes to withdraw funds from the master account, the funds must be transferred from a subledger account to the master account.

With a direct subledger, when a partner makes a withdrawal, funds are debited from the subledger. Funds cannot be withdrawn from the master account without transferring them first from the subledger to the master account.

Passthrough vs direct subledgers

Regardless of what type of subledger is used, the balance of the subledgers including the implicit subledger add up to the master account balance.

For direct subledgers, the implicit subledger is only dealing with funds coming in and out of the master. The implicit subledger doesn't float to balance the subledgers with the master as it does with passthrough subledgers. With direct, if the master account is funded directly, the implicit subledger will reflect that value. The rest of the value is in the other subledgers.

For instance, with direct, if a master account has 3 subledgers with $100 each and the implicit balance is at $0, the master will say $300. If $100 is deposited to the master account directly, then the implicit subledger will have a balance of $100, and the account will have a total balance of $400. If $400 was to then be withdrawn from the master, there would need to be a transfer of $300 ($100 from each subledger) to the master.

With a passthrough configuration, if the master account balance is $300, then $300 is withdrawn from the master without any corresponding debits from the subledgers.

With both passthrough and direct subledger configurations, all subledgers + implicit subledger = funds availability (master). The difference is that with a passthrough configuration, the implicit subledger and individual subledgers can be negative, whereas with direct configuration they cannot.

Conclusion

Passthrough is generally better for situations where a partner manages end-user deposits in multiple banks. For instance, a partner can manage customer funds at Cross River and two other banks, where the aggregate customer funds total $3 million, with 10 customers. With passthrough subledgers, there is no need for partners to specify which customer dollar is held in which subledger. With direct subledgers, it's more tricky, because partners need to not only consider how much their deposit balance is at Cross River in general, but how much they have allocated for each of their customers in subledgers.

A direct ledger system can be more difficult to manage, because to withdraw funds from the master account the funds must be transferred from its subledger accounts to the implicit master account. For this reason, it is recommended to consider the use case and configure in a way that will best serve your needs.

Passthrough	Key Features	Direct
Y	Instantaneous rollup of transactions to the master account balance	Y
Y	Fully routable on all payment rails (i.e. ACH, Wires, RTP, etc.)	Y
Y	Supports webhooks on subledger events (such as inbound transactions)	Y
Y	Reflects balance of transactions cleared against account	Y
Y	Subledger can carry a negative balance	N
Y	Will allow transactions to clear against subledger when balance < $0 (as long as the master account has sufficient funds)	N
N	Best for model where funds availability is managed at the subledger level	Y
Y	Best for model where funds availability is managed outside of subledger with need for visibility and efficiency of transactional flows with co-mingled funds	N

General ledgers

General ledger (GL) accounts are internal accounts used for accounting purposes for internal reconciliation. COS processes each transaction through a GL account. A GL account must reconcile to zero (0) at the end of the day.

Although GL accounts are meant for Cross River partner reconciliation. There are times when you might want to open a GL account for external purposes. For example, if you handle payroll deposits for a customer, you may need to overdraw on an account to be able to send salaries to the customers before the employer's funds have been deposited into your account.

2.4. Open an account

For any kind of money movement, you need a bank account. Your relationship manager helps you define exactly what types of accounts meet your needs. Each different type of account you define is also called a product type in Cross River COS. You use the product type as a basis to create individual accounts. Think of it as a template or configuration package.

Requirements to open an account

- Product ID: Cross River-provided unique product ID that references the defined product type of the new account. See **Account types and rules** for more details on how you get a product ID before opening an account.
- Customer ID: The system returns this unique ID when you **create a customer record**.
- Address and phone number: The account holder must have at least one address and phone number in their customer record.
- OFAC status: the account holder OFAC status must be *Clear*.
- Customer classification: The customer and product classifications must match. For example, only business customers can be added to a business product.

See our tutorial for **opening a deposit account** and our **API**.

Account types and rules

Cross River provides many different account types and solutions through a suite of products. You work with us to identify the types of accounts relevant to your business needs. For type of account type you want we create a template called a product type. The product type has a unique *product ID* you use to create specific accounts, so it has to be supplied before you can open an account.

Use cases by account type

When integrating with our platform, choosing the right account type is critical to support your product functionality, compliance, and reporting needs. These short descriptions can

help you determine which account structure is best suited for your use case.

Checking and savings accounts

Description

Standard accounts that can hold customer funds, support money movement (ACH, wires, card transactions), and accrue interest (savings).

Best for

- Holding end-user deposits (e.g., neobank checking/savings)
- Supporting card spend and ACH transfers
- Offering interest-bearing savings accounts
- General-purpose custodial accounts

Example use cases

- Neobanks offering everyday consumer banking
- Direct deposit accounts
- Spend-and-save features

Learn more about **Checking and savings**.

Subledgers

Description

Virtual accounts that sit under a parent account. These do not exist at the bank ledger level but are used for internal balance tracking and reconciliation.

Best for

- Tracking user balances without opening a full bank account for each user
- Managing program-level treasury with per-user accounting
- Supporting use cases that don't require individual account numbers

Example use cases

- Wallets or prepaid balances
- Marketplace payouts or platform credit tracking
- Funds segregation within a pooled account

Learn more about **subledgers**.

Certificates of Deposit (CDs)

Description
Time-bound deposit accounts that offer a fixed interest rate. Funds are locked for a specified term.

Best for

- Users or businesses looking to earn a higher yield on idle funds
- Offering fixed-term savings products
- Treasury management for long-term cash holdings

Example use cases

- Consumer savings products with higher yield and term commitments
- Business treasury optimization

Learn more about **Certificates of Deposit**.

FBO (For Benefit Of) Accounts

Description:
Legal accounts opened by a fintech or platform, holding funds **for the benefit of** end users. Often used to maintain compliance with custodial fund regulations.

Best for

- Programs where the fintech or platform acts as custodian

- Regulatory separation of end-user funds
- Platforms needing to manage funds on behalf of many users without full KYC at the bank level

Example use cases

- Investment apps holding funds before allocation
- Gig platforms managing worker payments
- Crowdfunding or pooled escrow solutions

Learn more about **For benefit of (FBO) accounts**.

Rules

When you create your product types, you also define the configuration of what that account can do. For example, you define which **payment methods** the account can use and any payment limits for for a given payment method.

Deposit, withdraw, transfer

Deposits, withdrawals and transfers can be understood as being part of a transaction. In the Cross River operating system (COS) a transaction always involves a debit of one account (withdrawal) and a credit of another account (deposit), completed as a single atomic (simple) operation. Both sides of the transaction must happen for it to be considered completed. If, for example, the debit account does not have enough money to cover the transaction, the credit operation will not complete.

This makes sure that accounts are never out of balance and makes tracking where the money is much easier.

Once you've opened an account, use our payment APIs to send and receive funds between banks via **ACH**, **Instant Payments**, **Wires**, **International payments** or **Checks**. You can also use Instant Payments to tranfer funds within Cross River, with our **CRNow** option.

Account activity record

When a deposit, withdrawal, or transfer is made, we update the account to show this activity. We call this an activity record. This ensures that the account balance is accurate and reflects all the money going in and out of the account. It's similar to updating a digital ledger to keep track of all your financial transactions.

A newly created activity record has a *Pending* status until the posting day closes during the nightly settlement process. At this point, COS reorders the account activity records for statement purposes as described in your deposit account agreement. COS assigns a posting sequence number to each account activity record and changes the status to *Posted*. Once this is done, the activity is considered final and is not updated further.

> The order of account activity presented on a given date is not necessarily the order that each activity happened.

For each deposit account Cross River provides a monthly account statement. The statement contains all posted activity inclusive of the period start and end dates. The statement lists details according to the posting sequence of the account activity records. The document is retrievable as a PDF file.

Post types

In banking, a "post" refers to the process of recording a transaction in your account. There are several types of posts.

Memo posts

Think of a *memo post* as a temporary transaction. It tells the system something about the account—usually that some money will be coming in or going out.

A memo post is applied only to one side of a transaction and may or may not affect the account available balance. A memo post might stay on an account for many days. When it expires it is removed and, as applicable, the information in it becomes a posted transaction activity record.

For example, when you pay for gas with a debit card, COS puts a memo post on the account until the transaction completes. When the charge is captured and finalized, the memo post is removed and converted to a posted transaction activity record, showing the actual amount spent at the gas station.

An example of an "informational" memo post is with an inbound ACH credit. When Cross River receives an instruction from the Federal Reserve about an inbound ACH credit, we put a memo post on the account right away even though the effective date of the actual transfer might not be until the following day. The job of this memo post is to tell the account holder that there is going to be some change to their account in the future.

You can **register** for account management **webhook events**, including memo post webhook events. A webhook event tells you the activity status.

Posted transactions

Availability schedules

Certain credit transactions post with an *availability schedule*. These schedules give instructions about the amount of the full transaction that will be available on each day after posting.

These days until all the funds are available are also called the *clearing window*. The schedule appears on both the transaction and account activity records. It appears as a array (list) of dollar amounts, where each element of the array represents a posting day.

For example, the following illustrates how an ACH pull transaction may look with a 2-day clearing schedule (lines 15-18):

Availability Schedule Example


```
{
      "id": "e25ff690-be93-469f-9915-aa9a011defbf",
      "transactionId": "c93978d3-3451-40a0-be01-aa9a011dec98",
      "traceNumber": "T192111321017433A",
      "status": "Pending",
      "transactionType": "Credit",
      "activityType": "External",
      "source": "Transaction",
      "masterAccountNumber": "2058112745",
      "subAccountNumber": "2058112745",
      "transactionCode": "Outgoing ACH",
      "description": "TEST ACH PULL PPD ID: 123 REF: A211P3HVG7KI",
      "rail": "Ach",
      "railId": "payment/eec1b00b-fcaf-4e91-a894-aa9a011d9366",
      "schedule": [
        0,
        0,
        2700
      ],
      "isClearing": true,
      "amount": 2700,
      "productId": "3c6d97f2-2666-44d9-b25d-a9a800d6c59d",
      "partnerId": "d6b4c661-b38a-46a3-8963-a9a40131eacf",
      "postingDate": "2019-07-30T00:00:00-04:00",
      "executedAt": "2019-07-30T13:21:01-04:00",
      "createdAt": "2019-07-30T13:21:03-04:00",
      "lastModifiedAt": "2019-07-30T13:21:03.7929693-04:00"
    }
```


The ACH payment is for $27.00 total. The schedule attribute shows that $0 will be available immediately on day 1 and that also nothing will be available on day 2. On day 3 the full $27.00 is available. Items such as checks generally make partial funds available each day while the item clears.

> Money amounts in API calls and responses are written without a decimal point between the dollars and the cents.

In addition to the `schedule` attribute, for convenience the response includes an `isClearing` flag to show whether the item is still within the clearing window. In the above example, the flag switches to false on 8/1.

The schedule is enforced via one or more account holds relating to the item in clearing, correlated by the `railId` field. For example, the schedule above would be held as follows:

Hold Example

```
{
      "id": "866d8cde-e7cf-4bf3-897b-aa9a014d9b08",
      "masterAccountNumber": "2058112745",
      "subAccountNumber": "2058112745",
      "status": "Active",
      "reason": "Availability",
      "rail": "Ach",
      "railId": "payment/eec1b00b-fcaf-4e91-a894-aa9a011d9366",
      "amount": 2700,
      "createdAt": "2019-07-30T16:14:37-04:00",
      "expiresAt": "2019-07-31T00:00:00-04:00",
      "partnerId": "d6b4c661-b38a-46a3-8963-a9a40131eacf",
      "lastModifiedAt": "2019-07-30T16:14:37.287877-04:00"
    }
```

Note that the hold expires at the close of business on 7/31, thus making the $27.00 available starting on the 8/1 posting day.

Interest accrual is not affected by availability schedules. Accrual will continue to start on the date of the credit regardless of any holds which are in place.

Restrictions

Restrictions provide a mechanism to block certain transaction activity on an account

For example, if you wanted to block all ACH activity on an account, you could add a restriction which only applies to the ACH rail. If you then wanted to entirely freeze an account, where all activity was blocked, you could add a restriction which applies to all rails.

Partner vs. internal restrictions

As the name suggests, a Partner restriction is one that is applied by a Partner user. It is visible to all users and can be modified by both Partner and Internal users. By contrast an Internal restriction is one that is visible to Partner users, but cannot be modified or removed by them. For example, an internal Operations Team member may apply an Internal account freeze if they do not want it to be modified by the Partner.

This division between Partner restrictions and Internal restrictions also applies to holds.

> **IMPORTANT**
> Should I display restrictions to my customers?
> Restrictions and holds may be sensitive in nature. Your Integration Manager (IM) will assist you with your questions on how best to handle customer support issues arising from restrictions or holds.

Restriction policies

Restriction policies are managed at the product level and define the types of transactions allowed for the program as a whole. Account level restrictions can be applied in addition to these. These policies are not managed by the partner and are configured by the Integration Team for your product.

Creating restrictions

Account restrictions work by defining a set of criteria in which the restriction applies to. For example, if we wanted to block all ACH debits over $100 to an account we would create a restriction like this:

Curl

```
POST https://sandbox.crbcos.com/core/v1/dda/accounts/200012345/restrictions
{
  "subAccountNumber": null,
  "appliesTo": "All",
  "rail": "Ach",
  "transactionType": "Debit",
  "otherAccountType": null,
  "amountThreshold": 10000,
  "transactionCode": null,
  "reason": "Restricted",
  "notes": "Optional notes here"
}
```

Null values indicate a wildcard that any value would match.

Posting exceptions

A posting exception is like an account activity record, except it only happens when one side of a transaction fails authorization. For example, if the debit side of a transaction fails due to insufficient funds in the account, COS creates a posting exception record on that account. COS does not put a posting exception on the credit side of the transaction because it did not fail in this example.

Update details

Sometimes even after you open a bank account, you need to add or update details of the account holder **customer record**. You can update the account holder address, phone number and email. You can also add details to the customer record about an ID document for that account holder.

For every record, you can have multiple addresses and so forth, but you designate one of each as the primary detail. For example you have a primary address and phone number.

Among other purposes, these details are important for identity verification to fight fraud and money laundering.

See our customer record management **APIs** and **tutorials** to learn how to update or add this information.

2.6. Accounts features

Cross River accounts give you a number of special features and functionalties:

- **Interest rates**: What we offer and how to apply interest rates to savings accounts
- **Deposit sweeps**: Our feature to increase the scope of deposit insurance for your customers
- **Statements and tax forms**: Download monthly statements and 1099s

2.6.1. Interest rates

Cross River offers both flat and tiered interest rates using a compounding method. Interest accrues daily and is posted monthly, on the last day of the month.

When you set up an interest-bearing account **product type**, the interest rate profile is part of a product's core setup. We assign a default interest rate profile when configuring a product type, even if it's the only rate card added. This ensures the interest rate applies to any account opened for that product type. Adding multiple rate profiles gives you the flexibility to change interest rates as needed.

Since rate profiles are part of the core setup, only the Cross River operations team can configure them. Once configured, we provide you with the `rateCard ID`s. It's important to store all your rate card IDs in case you need to change a rate profile on an account.

> **IMPORTANT**
> There is no endpoint or attribute that allows you to view information on the rate profile created for your products. You must store that information as well as your rateCard IDs we provide.

Tiered interest rate accounts

Tiered interest rate accounts are an excellent way for you to offer customers higher interest rates as their balances grow. With these accounts, the interest rate can be configured to increase with higher balances, providing a better return on their money.

Here's how it works: The rate card is divided into different tiers, each with its own interest rate. For example, the first tier might offer 1% interest on balances up to $1,000. If the balance exceeds $1,000, the next tier might offer 2% interest on the amount above $1,000.

This setup encourages customers to save more since higher balances earn higher interest rates, helping customers grow their savings more effectively. It also provides a clear and structured way for you to offer competitive savings products. By using tiered interest, you can attract and retain customers who are looking for better returns on their deposits.

Tiered interest rate limitations

- Not available for customers using savings account sweeps functionality
- Activity-based actions (for example, make one deposit and two transactions, your rate will be x%) are not supported
- Overall balance triggers (for example, the balance is >$10,000, pay 2% on entire balance, otherwise pay $0) are not supported

Configure your rate profile

When working with the Cross River Operations Team to create your rate profiles, you must provide the following information, regardless of whether or not you want to create tiered interest:

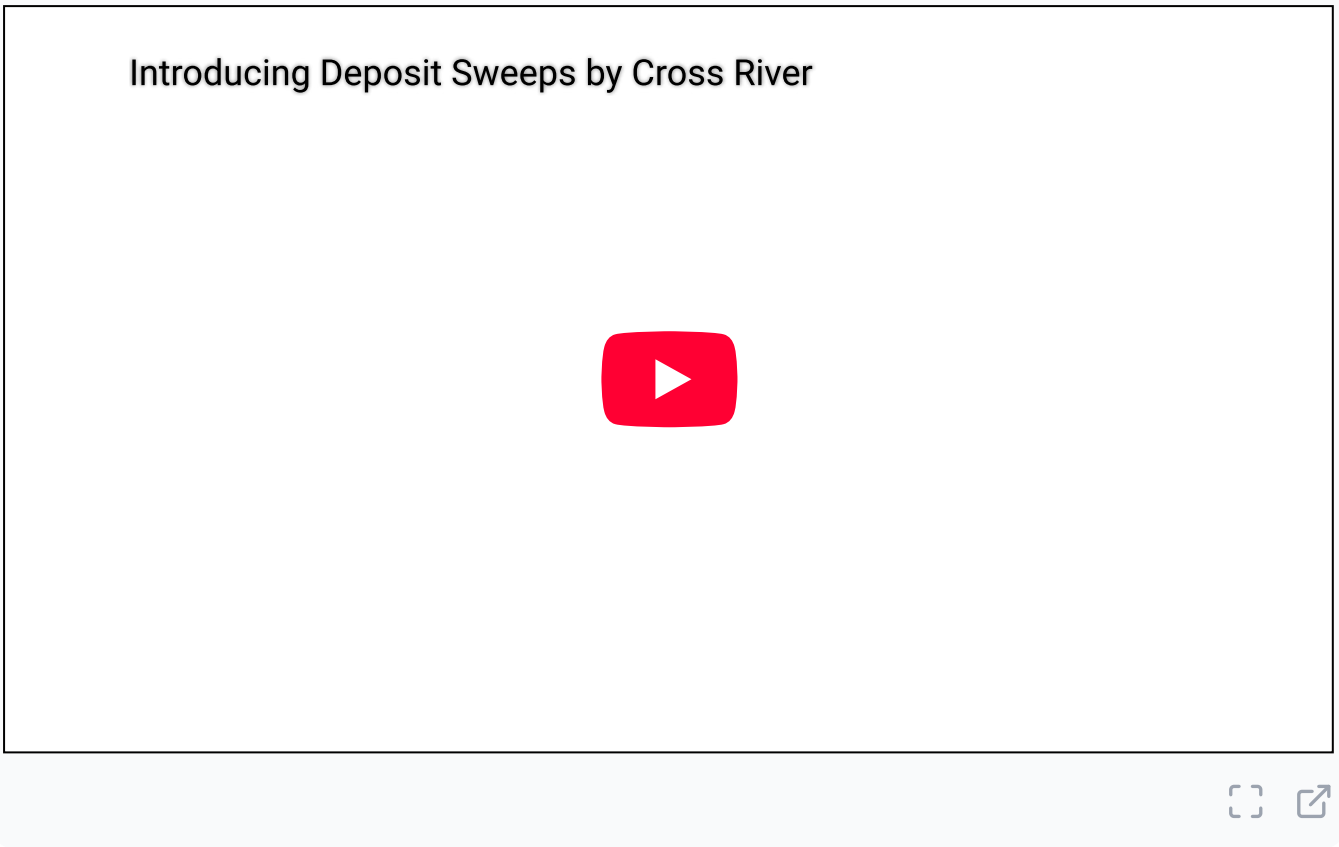
Field	Description
Tier	The tier number within the rate card. This is automatically assigned when a new interest rate card is created and when multiple tiers are added.
Start	The date the tier comes into effect. Optional field. By default, takes effect the date the rate card is created.
Rate Method	How the interest rate is calculated
Partner %	The percentage of the Effective % paid by you
Min Balance	The minimum balance required for the rate tier
End	The date when the tier is no longer active. Optional field. By default, no value.
Effective %	The total effective interest rate provided by the tier

Link a rate card to an account

Linking a rate card to an account is an additional step in the onboarding process. To link a rate card to an account:

1. Create a customer (**POST /v1/cm/customers**)
2. Create an account for that customer (**POST /v1/dda/accounts**)
3. Call `PUT /v1/dda/accounts/{accountNumber}/rateCard` to attach an existing rateCard ID to the account. The rate card is defined in the `rateCard` attribute.

2.6.2. Deposit sweeps



Deposit Sweeps APIs

Cross River partners with IntraFi to offer you deposit sweeps. IntraFi connects Cross River to a large network of banks and also handles the complexity of managing deposit placements across the network of institutions. Through our partnership with Intrafi, you can increase FDIC insurance coverage for your customers' deposits up to $170M per depositor (in each insurable capacity).

Why you'd be a great candidate

- You want to offer your customers an insurance limit of up to $170 million per tax ID.
- Your customers deposit accounts have cash balances of over $250,000.

Prerequisites

To implement deposit sweeps, you need to:

1. **Register as a Cross River partner:** Required for new partners.
2. **Request sandbox access:** You'll need to:
    - Have a deposit product
    - In addition to your regular configuration, be sure to include IntraFi Sweeps
3. **Sign the service agreement:**
    - For a new partner, ensure it covers deposit products.
    - For an existing partner, covers deposit IntraFi Sweeps.
4. **Complete sandbox certification:** Ensure all testing is validated and certified.
5. **Obtain production access:** Perform production testing to finalize integration.
6. **Enable deposit IntraFi Sweeps capabilities:** Activate services for your account product on Cross River.

> Before you offer IntraFi Sweeps to your customers, they must accept the IntraFi deposit placement agreement.

Key features

Once you've added IntraFi Sweeps to your deposit product, our APIs let you offer your customers the ability to:

- View deposit placements and balances. See the breakdown of where funds are and how much is at each institution.
- Settlement happens every day. Cross River sends records to IntraFi of what needs to be swept or returned and IntraFi accepts those records.
- At the end of day, either Cross River sends a settlement wire of funds to IntraFi or Intrafi return funds to Cross River.
- **Opt out** of a particular bank. We recommend customers do this if they already have funds at a specific institution.

Sample flow

In our example, you use Cross River as your partner bank. You offer a deposit product enabled for IntraFi sweeps. The flow looks like this:

1. Your customer opens an account and deposits $750K.
2. This $750K now resides in a Cross River Account.
3. Since this deposit is enabled for sweeps, anything in excess of $250K will get picked up for sweeps through the IntraFi network.
4. Cross River wired $500K to IntraFi.
5. IntraFi handles the relationship with a network of banks. IntraFi sweeps $250K into two FDIC compliant banks.
6. Your customer is insured for the entire available balance of $750K.
7. $250K remains at Cross River, but the available balance remains at $750K.
8. Your customer can withdraw from their full balance at any time.

Reasons to choose Cross River

Once we enable the deposit sweeps add-on for your deposit product, *any* of your customer accounts with funds in excess of the FDIC insurance limit are picked up for sweeps and funds will be swept to the IntraFi network.

- **Account balance always visible**: The total account balance remains visible as if all the funds are in a single account.
- **Interest rate stability:** The interest rate on the various accounts remains same as offered in your Cross River deposit account product.
- **Withdraw funds at any time:** The available balance includes the swept balance and your customer can withdraw funds at anytime.

FAQ

What happens if you have enabled a partner for Sweeps and then they open 1000 accounts, is that more work for CR?

No, because the product is what is enabled for Sweeps so every deposit account under that product will be picked up for Sweeps.

Can a customer choose to opt out for a specific bank or account?

Yes, they can be added to an exclusion list.

How do I enable IntraFi Sweeps?

It's a simple product configuration change performed by Cross River.

What happens to the deposit product interest rate if I enable Sweeps?

The interest rate stays the same. IntraFi honors the interest rate on the account with Cross River, even if the rate is different at the other banks.

Next steps

- Have a look at our sweeps **APIs**.
- **Get started** with Cross River.
- Any questions? Contact Cross River **support**.

2.6.3. Statements and tax forms

Statements

For each deposit account CR provides a monthly account statement. The statement contains all posted activity inclusive of the period start and end dates. The statement lists details according to the posting sequence of the account activity records. The document is retrievable as a PDF file through COS Explorer or using the `GET /v1/dda/accounts/{accountNumber}/statements/{statementId}/documents` endpoint.

Tax document retrieval

Cross River enables you to get PDFs of tax documents such as IRS Form 1099. When your customer wants to see a tax document, use our APIs to retrieve and display the PDF in your app. Use COS Explorer if you want to retrieve a single document directly.

Your relationship manager lets you know when a specific type of tax document is available for download.

Tax document retrieval uses these API endpoints:

- **Get tax documents**
- **Get a specific tax document**

> **IMPORTANT**
> Only pull documents as needed. If you want to pull a large number of documents, speak to your relationship manager.

> The IRS requires Cross River to provide account holders with an IRS Form 1099 for earnings equal to or greater than $10.00, per taxpayer identification number (TIN). Should the customer have more than one account, the system generates a 1099 for *each* account if the earnings of *all* accounts total $10.00 or more.

2.7. COS Explorer

Create and manage your customer deposit accounts from the **Deposit Accounts** screen in the **Core** tab of COS Explorer.

Deposit account details

In the **Deposit Accounts** screen, click an entry to view a selected account. A screen displays with comprehensive information about the account.

You can see the current balance and available balance of the deposit account

View balance history

To view the balance history, click for either the current or available and enter the date range for the information you want to see. Click **Search** and the records appear in the dialog box. The records include the following:

Field	Description
TYPE	The type of entry
CURRENT	Amount shown in the account at the defined entry time. An amount in parentheses means a debit and without parentheses means a credit.
RUNNING CURRENT	The running balance amount shown in the account at the defined entry time including all transactions whether or not complete at that time
AVAILABLE	Amount available for use at the defined time of the entry. An amount in parentheses means a debit and without parentheses means a credit.
RUNNING AVAILABLE	The running balance of available funds at the defined time of the entry. A negative amount means a debit and a positive amount a credit.

Deposit account details area

The following information displays in the **Deposit Account Details** area:

Field	Description
Acct #	Account number
Classification	Account classification (Personal / Business)
Status	Account status
Product Type	Product type (General Ledger/Checking/Savings/Time Deposit/Wallet)
Opened	Date account was opened
Last Maintenance	Date of last maintenance on account
Last Contacted	Date of last contact with customer to verify account
Partner	The partner under which the customer has been created**Note:** Click the partner name to see the partner details view
Statement Address	Full address associated with account
Credit Limit	The maximum amount the deposit account can be overdrawn
Available Credit Limit	Available funds in the deposit account
Overdraft Enabled	Allows an account to be overdrawn according to partner-customer agreement/ terms
Overdraft Account	A separate account with a maximum limit of money that can be drawn to fund the deposit account
Overdraft Threshold	The desired end-of-day balance in the account

Field	Description
Title	Account title
Client Identifier	Use this attribute to add your own unique identifying string to a payment call or COS record. This attribute is useful for **idempotency handling**.
Ledger Type	Account ledger type. **Passthrough** (default) or **Direct**
Rate Card	Preconfigured template assigned to account, containing interest rates **Note:** To view details on a rate card, click the rate card name
Last Transacted	Date of last transaction on account
Product	Product account is associated with
Contact Info	Contact information for the customer

Click the 3-dot menu for options on editing the deposit account details as described below.

The new deposit account details display in the **Deposit Account Details** area.

What action?	How	Steps
Change Account Title	Click the 3-dot menu and select **Change Account Title**.	Edit the account title field and click **Submit**
Report Customer Contact	Click the 3-dot menu and select **Report Customer Contact**.	Edit the customer contact fields, the internal only note, partner note and click **Confirm**
Change Statement Address	Click the 3-dot menu and select **Change Statement Address**.	Edit the statement address fields and click **Save Changes**
Change Status	Click the 3-dot menu and select **Change Status**.	Edit the account status fields and click **Save Changes**
Change Account Credit Limit	Click the 3-dot menu and select **Change Account Credit Limit**.	Edit the account credit limit and click **Submit**
Change Overdraft Funding	Click the 3-dot menu and select **Change Overdraft Funding**.	Edit the overdraft funding fields and click **Save Changes**

Customer relationships

The **Customer Relationships** area provides information about the customers with relationships to the account.

Column	Description
CUSTOMER	Customer that account is related to
RELATIONSHIP	Customer relationship to account
TAX OWNER	Indicates whether customer is a tax owner
DATE ADDED	Date relationship was added to account

1. Click the 3-dot menu, **Add Customer Relationship** to add a new customer relationship for the deposit account.
2. Select the **Customer** and **Relationship**.

3. Click Submit
   Click **x** to remove a customer.

> The account always has an assigned primary. If an additional person is added to the account and the primary then dies the additional person does not become a primary. The account would have to be closed and a new account opened.

Click a customer to view their detailed information.

Account activity tab

The **Account Activity** tab displays the **Current and Available** balance and information on the deposit account activity.

Column	Description
DATE	Activity date
SUB LEDGER	Number of deposit sub ledger account on which activity took place
DESCRIPTION	Activity description
CODE	Transaction code of activity
AMOUNT	Dollar amount of activity
BALANCE	Account balance following activity

Sometimes you need to indicate that funds will be debited from or credited to an account. Create a memo post for that purpose.

1. Click **Create Memo Post**.
2. Enter memo post details and click **Save.**

You can filter the information to show specific account activity. Enter the parameters that interest you and click the filter icon.

Field	Description
Date From	Earliest activity date
To	Latest activity date
Amount From	Minimum dollar amount to filter on
To	Maximum dollar amount to filter on
Type	Internal/ External
Code	Type of transaction, including account transfers, incoming and outgoing ACH transactions, and any ACH NOC
Currency	Currency type. Choose from the drop-down list.

Click a transaction in the list to view detailed information about it. For more information on viewing deposit account transactions, see **Working with Deposit sub ledger** accounts.

Posting exceptions tab

The **Posting Exceptions** tab displays transactions that were not posted within a set date period for the account.

Field	Description
DATE	Posting exception date
TRACE #	Unique core transaction trace number of posting exception
SUB LEDGER	Number of sub ledger account on which exception occurred
DESCRIPTION	Description of transaction for which exception occurred
CODE	Code of activity on which exception occurred
REJECTION	Rejection reason code
MESSAGE	Rejection reason
AMOUNT	Dollar amount of activity on which exception occurred

You can filter the displayed information. Enter the parameters that interest you and click the filter icon.

Click an item in the list to view detailed information about it.

Debit cards tab

> The Debit Cards feature is only supported for specific partners prior to 2020.

The **Debit Cards** tab displays information about the card and cardholder.

This is intended for creating and managing **only** customer debit cards using VisaDirect and MastercardSend (McSend) as the payment processor.

Field	Description
LAST FOUR	Last 4 digits of card
CARDHOLDER	Name of cardholder
STATUS	Active/Inactive
OPENED	Date card created
PRIMARY	Whether primary card
ADMIN LOCKED	Locked by admin

To create a new debit card:

1. Click the 3-dot menu, select **Add New Card** and enter card details.
2. Click **Submit** to save.

Statements tab

The **Statements** tab displays information about a deposit account for a defined period (this must have a start and stop date that has passed).

> **IMPORTANT**
> The statement type you can pull is dictated by the **roles** that you are assigned on configuration. Your Explorer admins are responsible for making sure that users are properly configured with the appropriate roles and associated limits. Statements can only be pulled for a time period that is in the past.

Field	Description
STATUS	Statement status
PERIOD	Time period covered in statement
ENTRIES	Number of entries in statement
BEGIN BALANCE	Balance at beginning of time period covered in statement
END BALANCE	Balance at end of time period covered in statement

Click the 3-dot menu, and select **Generate Statement** to generate a statement for a defined time period. Click the download icon to download the generated statements.

Currently, statements are generated and can be downloaded monthly. You can generate a statement on demand however the following rules apply:

- The statement is created for a range of account activity that has not already been included in another statement. So if a statement already exists for the month of January then another statement that includes both January and February cannot be created.
- The statement only includes activity which has been settled. This means all account activity which has not been hard posted during the nightly settlement process will not be eligible.

Click an entry to display details for that month's statement. A screen opens showing **Statement Details** and **Statement Entries**.

Statement details

Field	Description
Account Number	Account number
Statement Address	-First Name -Middle Name -Last Name -Street Address 1-3 -District -City -ZIP code -Country
Status	Statement status
Period	Statement period
Beginning Balance	Balance at the beginning of the time period covered in the statement
Interest Accrued	Total amount of interest accrued on the account
Interest Paid	Total amount of interest paid on account
Withdrawals	
Debits	Total dollar amount of debits
Created	Date and timestamp of statement
Product	Name of product
Ending Balance	
Interest Accrued YTD	Interest incurred year to date
Interest Paid YTD	Amount of interest paid on the account from the beginning of the current year to the current date
Annual Percentage Yield Earned	Amount of compound interest earned on account in the year
Entries	Number of entries in the statement

Field	Description
Deposits	Total dollar amount of deposits over the statement period
Credits	Total dollar amount of credits over the statement period
Completed	Date and time that the statement was completed

Statement entries

Field	Description
DATE	Activity date
SUB LEDGER	N/A
CODE	Transaction code
DESCRIPTION	Transaction description
AMOUNT	Transaction amount
BALANCE	Balance following the transaction

Statement tasks

From the **Statement Tasks** area in the **Statement Details** screen you can perform several statement tasks. These include:

- Recall
- Mark delivered
- Download document

From the **Account Statements** area you can go to other monthly statements. Click on the plus sign by each calendar year to view more.

Holds tab

In the **Holds** tab, you can search for and view holds on a deposit account using these search filters:

Field	Description
Rail	Rail type
Rail Id	Rail ID
Reason	Hold reason

Deposit account hold information displays as follows:

Column	Description
REASON	Hold reason
CREATED	Hold creation date
EXP. DATE	Hold expiration date
NOTES	Notes entered on hold
RAIL	Rail type
TYPE	Hold type
AMOUNT	Dollar amount of hold

Remove a hold from the list of holds by clicking **x**.

To place a hold on an account:

1. Click the 3-dot menu. Select **Place Hold** and fill in the fields.
2. Click **Save**.

Restrictions tab

The **Restrictions** tab displays details of restrictions placed on a deposit account.

To remove a restriction from the list of restrictions, click **x**.

Column	Description
APPLIES TO	The account(s) the restriction applies to
RAIL	Rail type
OTHER ACCT	If there are any other accounts that are impacted by the restriction. Restrict transaction according to specific account type.
TRANS TYPE	Transaction type of restriction - Debit -Credit
TRANS CODE	Transaction code of restriction
THRESHOLD	Dollar amount threshold of restriction
TYPE	Restriction Type -Partner -Internal
EFFECTIVE	
REASON	Restriction reason
NOTES	Notes entered on restriction

To add a restriction on the deposit account:

1. Click the 3-dot menu. Select **Add Restriction** and fill in the fields.
2. Click **Save**.

Stop payments tab

The **Stop Payments** tab displays information on payments the account holder has requested be stopped.

Click a row in the list of stop payments to view details.

Field	Description
RAIL	Rail type of stop payment
STATUS	Status of stop payment
MIN	Minimum dollar amount of stop payment
MAX	Maximum dollar amount of stop payment
EXPIRATION	Expiration date of stop payment
NOTES	Notes entered on stop payment

To stop a payment:

1. Click the 3-dot menu, select **Stop Payment**, and fill in the fields as necessary.
2. Click **Place Stop Payment**.

Account snapshot

The **Account Snapshot** area displays summary information by Current Month, year-to-date (YTD) and Life Time.

Field	Description
Transaction Count	Number of transactions for given time period
Low Balance	Lowest balance in time period
High Balance	Highest balance in time period
Withdrawal Total	Total dollar amount of withdrawals over time period
Deposit Total	Total dollar amount of deposits over time period
Debit Total	Total dollar amount of debits over period
Credit Total	Total dollar amount of credits over period
Last Debit	Dollar amount of last debit in period
Last Debit At	Date of last debit in the time period
Last Credit	Dollar amount of last credit in time period
Last Credit At	Date of last credit in time period
Last Interest	Dollar amount of last interest earned in time period
Last Interest At	Date of last interest earned in time period
Interest Total	Total interest earned over time period
Overdraft Count	Number of overdrafts during time period
Overdraft Days	Number of overdraft days during time period
Uncollected Count	Number of transactions rejected due to uncollected funds

Deposit account tasks

From the screen detailing a specific account you can perform these actions on your account:

Deposit funds

See Posting transactions.

Withdraw funds

See Posting transactions.

Close account

> To close an account the balance must be zero.

1. Click **Close Account** and fill fields.
2. Click **Confirm**.

Open sub ledger

See Open a deposit sub ledger account.

View subledger

The subledger accounts are listed in the **Deposit Accounts Sub Ledgers** search screen. For more information see **subledgers**.

View interest

Displays details of the breakdown of interest paid on an account.
Click the filter icon to toggle the search fields, and enter relevant search parameters.
Click **Search**.

- **Accruals Tab**

Field	Description
PERIOD START	Start date of interest accrual period
PERIOD END	End date of interest accrual period
AMOUNT	Amount of interest accrued
ANNUAL RATE	Annual interest rate
PRINCIPAL	Principal amount of interest
PAID	Indicates whether interest was paid
PAID ON	Date of interest payment
RATE CARD	Pre-configured template assigned to account containing interest rates
TIER	Interest rate tier as indicated in assigned rate card

- **Payments Tab**

Field	DESCRIPTION
TRANSACTION	Contains a link to the Core transaction details, viewable in the Transaction Information screen.
PAID ON	Interest payment date
AMOUNT	Interest payment amount
UNPOSTED AMOUNT	Unposted interest payment amount
ACCRUAL AMOUNT	Amount of interest accrued
CARRY OVER AMOUNT	Amount of carry over interest
LIFETIME PAYMENT	Total amount of lifetime interest payment
POSTED	Indicates whether interest payment was posted
POSTED ON	Interest payment posting date

View tax documents

A record of the tax documents associated with the deposit account displays here. Click **View Tax Documents**.

Field	Description
TAX YEAR	Tax year for the document
STATUS	Tax document status
ACCT #	Account number associated with tax document
CUSTOMER	Customer name on account
TYPE	Tax document type
CORRECTION	Indicates whether a correction was made to tax document

View settlements

Click **View Settlements** to display a list of deposit account settlements within a date period.

Field	Description
DATE	Settlement date
START	Number of transactions settled over lifetime of account up to date in DATE field
END	Confirmed number of transactions settled over lifetime of account up to date in DATE field
COMPLETED	Date and time of settlement completion

Click a settlement in the list to view the settlement details.

Field	Description
Uncollected	Uncollected funds on settlement
Overdrafts	Overdrafts on settlement
Transactions	Transactions on settlement
Debits	Total dollar amount of debits on settlement
Last Debit	Amount of last debit on settlement
Credit	Total dollar amount of credits on settlement
Last Credit	Amount of last credit on settlement
Deposits	Total dollar amount of deposits on settlement
Withdrawals	Total dollar amount of withdrawals on settlement
Opening Balance	Opening balance on the account as of settlement
Closing Balance	Closing balance on the account as of settlement
Low Balance	Lowest balance on the account as of settlement
High Balance	Highest balance on the account as of settlement
Interest Total	Total interest earned on account

To search the settlements by a selected date period, click the filter icon to toggle the search fields, and enter the starting and ending dates. Click **Search**.

View notes

You can see any notes that have been saved for the deposit account.

Field	Description
DATE	Note creation date
SUBJECT	Note subject
AUDIENCE	Note audience (Internal or Partner)
CREATED BY	COS user who created note
BODY	Note text

To add a note the account:

1. Click **Add Note** and fill in the fields in the **Create New Note**.
2. Click **Submit**. The new note appears in **View Notes**.

View tags

Click **View Tags** to display tag information.

Column	Description
TAG	Tag name
STATUS	Tag status
AUDIENCE	Tag audience (Internal or Partner)
DATE	Tag creation date
CREATED BY	Tag created by

Click a tag in the list to view the details of the tag.

To create a new tag:

1. Click **Add Tag** and fill in the fields.

2. Click **Submit**, the new tag appears in the **View Tags**.

Search for deposit accounts

1. Go to **Core**>**Deposit Accounts**.

2. **Search by Id** or click `Filter` and enter your search parameters:

Filter	Description
Partner	Partner name or select from list of partner names
Product	Product name
Acct #	Deposit account number
Title	Account title
Status	Account status
From	Search from date
To	Search until date
Client Identifier	Use this attribute to add your own unique identifying string to a payment call or COS record. This attribute is useful for **idempotency handling**.

3. Click **Search**. The search results display in the **Deposit Accounts** list. Each row in the list represents an individual deposit account.

Field	Description
ACCT #	Account number
STATUS	Account status
TITLE	Account title
PRODUCT	Account product
PARTNER	Partner to who account is associated
OPENED	Date account was opened
CURRENT BALANCE	Current balance of account
AVAILABLE BALANCE	Available balance of account

Open a deposit account

1. Click **Core**.
2. Click **Deposit Accounts**.
3. Click **Open Account**.
4. Fill in the fields.

Field	Description
Partner	Partner
Product	Product
Customer	Customer
Account Title	Account Title
Street [1-3]	Street [1-3]
City	City
State/ Province	State/ Province
Postal Code	Postal Code
Country Code	Country Code

3. Cards

Card Issuing APIs

Cross River delivers a flexible, scalable card issuing platform that enables partners to create debit, credit, and prepaid card programs. Our solutions are designed to meet a range of customer and business needs — from traditional deposit-linked debit cards to innovative prepaid products and various credit offerings. Partners can offer both physical and virtual cards and select the issuing model that best fits their operational and technical requirements.

What Cross River offers

Card product options

Cross River supports three primary card product types. Selecting the right type of card is a foundational step in building a compliant and scalable card program.

- **Debit cards**: Debit cards allow our partners' customers to access available funds directly through card network transactions. This card product is associated with funds in an underlying bank account. The customer's account is debited by the amount of the transaction when the card is used.
- **Credit cards**: Credit cards provide customers with access to a line of credit. Cards can be structured as secured (collateralized by customer deposits or other assets) or unsecured (based on creditworthiness). Interest can be charged on any outstanding balance carried over month to month, depending on specific program terms.
- **Prepaid cards**: A prepaid card is not linked to a traditional bank account. Instead, it allows a customer to spend money that has been loaded into a prepaid card account in advance. They offer flexibility for general-purpose spending, payroll, corporate-use cases, and mobile-first experiences.

Issuing models

Cross River offers two flexible operating models for card issuing.

- **Partner-managed**: This model provides Cross River partners with maximum involvement in how the card program operates.
- **Integrated issuing**: This model offers partners a Cross River-managed infrastructure for launching and managing card programs.

Program services

Cross River provides essential banking, compliance, and settlement services to support the full lifecycle of a card program.

- **BIN sponsorship**: Cross River sponsors the use of Bank Identification Numbers (Card BIN), enabling partners to offer cards without needing direct membership with Visa or Mastercard.
- **Compliance and risk management**: Cross River handles regulatory obligations including KYC onboarding, AML transaction monitoring, OFAC screening, and network compliance oversight.
- **Settlement and reporting** : Cross River manages daily settlement with card networks, financial reconciliation, and operational reporting.
- **Processor flexibility**: Partners may work with supported processors already integrated with Cross River or bring their own processor [1]. Integration models are designed to minimize technical overhead while meeting compliance and reporting requirements

> [1] This is subject to due diligence review and at in the sole discretion of Cross River

Find out more

- **Issuing models**
- **Card lifecycle**
- **Card transactions**
- **Reporting and compliance**

3.1. Issuing models

Cross River offers two flexible operating models for card issuing. Partners can select the approach that best matches their technical capabilities, operational preferences, and program goals.

Feature	Partner-managed	Integrated issuing
BIN sponsorship	CR managed	CR managed
Compliance	CR managed	CR managed
Network settlement and reporting	CR managed	CR managed
Processor integration	Partner	CR managed
Ledger management	Partner	CR managed
Transaction authorization	Partner or processor	CR managed
Card network connectivity	Partner	CR managed
Card lifecycle management	Partner or processor	Partner, using CR APIs
Customer experience and app development	Partner	Partner, using CR APIs

Partner-managed issuing

In a **partner-managed** model, Cross River partners maintain control over ledgering, authorization, and customer experience by working with a processor of their choosing. (insert footnote referenced above) Cross River provides BIN sponsorship, regulatory oversight, and manages settlement with card networks. This model is ideal for partners who already have established processor relationships and operational capabilities. It enables greater customization of authorization rules, ledger management, and customer-facing experiences while leveraging Cross River's banking infrastructure and compliance oversight

Integrated issuing

In an **integrated issuing** model, Cross River provides BIN sponsorship, acts as the ledger, manages transaction authorization, integrates directly with supported processors, and provides network and regulatory compliance oversight. This model simplifies setup, reduces technical burden, and enables faster time to market for our partners. It is designed for partners who want to focus primarily on delivering customer experience without managing banking infrastructure or complex back-office processes.

3.1.1. Partner-managed

In Cross River's partner-managed model, Cross River partners have maximum input into the card program.

- Partners use their own card processor and handle everything from ledgering to transaction approvals. They manage the entire customer experience while the bank performs program oversight.
- Card Management: Partners take care of card creation, activation, suspension, closure, and replacement using their processor systems within the construct agreed to by the bank.

What Cross River offers

Cross River acts as the issuing bank, providing BIN sponsorship, banking oversight, guidance to maintain best practices, compliance support, and network settlement. Cross River also provides regulatory expertise and operational support.

This model allows partners to tailor card programs while benefiting from Cross River's compliance oversight and expertise.

Cross River's role

As the issuing bank, Cross River provides:

- **BIN sponsorship:** As a Principal member, Cross River sponsors the use of its Bank Identification Numbers (BINs), enabling partners to offer cards without needing direct membership with Visa or Mastercard. Cross River has been assigned BINs that can be used to sponsor card programs that partners may extend to their customers.
- **Oversight:** Cross River ensures that card programs comply with Visa, Mastercard, and federal/state banking regulations from initial program evaluation and through ongoing oversight.
- **Network settlement and reporting**: Cross River monitors program activity through review of daily files, transaction monitoring **reports**, and exception handling

processes to maintain operational integrity. Cross River manages network settlement with Visa and Mastercard.

Partner's role

The partner is responsible for:

- **Processor relationship and integration:** Partners select and manage the card processor (insert footnote reference). The processor connects directly to the card networks (Visa, Mastercard) on behalf of the issuing BINs assigned by Cross River. Partners then manage the ongoing operational relationship with the processor and customer experience flows, while exchanging information with Cross River to support settlement and compliance oversight.
- **Ledger management:** Partners maintain a ledger of customer balances and transaction history, aligned with Cross River's compliance requirements.
- **Transaction authorization:** Partners manage real-time transaction approvals through the card processor, applying program-defined business rules, fraud controls, and transaction limits. Cross River works with partners with the goal of ensuring these rules are compliant with network standards and regulatory expectations.
- **Card network connectivity:** Partners are responsible for establishing and managing connectivity between their processor and the card network (Visa, Mastercard). This includes receiving authorization requests, submitting clearing files, and handling chargebacks and disputes.
- **Card lifecycle management:** Partners manage card creation, activation, suspension, closure, and replacement flows through the card processor systems, with Cross River providing guidance to maintain best practices and compliance. Refer to **partner-managed** for more details.

Find out more

- **Card lifecycle**
- **Card transactions**
- **Reporting and compliance**

3.1.2. Integrated issuing

Cards APIs

Integrated Issuing lets partners use Cross River's infrastructure to launch and manage card programs [1]. This model is perfect for partners who want a convenient way to offer cards without dealing with complex operations.

- Cross River acts as the issuing bank, keeps the records, and authorizes transactions. Cross River handles core banking, compliance, and network operations.
- Partners focus on customer experience and developing their applications

> [1] This is subject to due diligence review and at in the sole discretion of Cross River

What Cross River offers

As with the Partner-managed framework, Cross River acts as the issuing bank, providing BIN sponsorship, banking oversight, guidance to maintain best practices, compliance support, and network settlement. Cross River also provides regulatory expertise and operational support. In addition, Cross River provides the technology infrastructure for card processing, relieving the Partner of the need to find and integrate with a separate third party for this purpose.

Cross River's role

Banking and program oversight

As the issuing bank, Cross River provides:

- **BIN sponsorship:** As a Principal member, Cross River sponsors the use of its Bank Identification Numbers (BINs), enabling partners to issue cards without needing direct membership with Visa or Mastercard. Cross River has been assigned BINs that can be used to sponsor our partners' card programs that they extend to their customers.

- **Oversight:** Cross River ensures that card programs comply with Visa, Mastercard, and federal/state banking regulations through initial program evaluation and ongoing oversight.
- **Network settlement and reporting:** Cross River monitors program activity through review of daily files, transaction monitoring **reports**, and exception handling processes to maintain operational integrity. Cross River manages network settlement with Visa and Mastercard.

Technical infrastructure

Cross River powers the technical foundation for card programs:

- **Processor integration:** Cross River integrates with supported processors to relay authorization requests and transaction messages between card networks and Cross River Operating System (COS).
- **Ledger management:** COS maintains ledgers for customer accounts, card balances, and transaction history.
- **Transaction authorization:** COS applies real-time decisioning based on card status, available balance, program rules, and fraud controls.
- **Card network connectivity:** Cross River connects with Visa and Mastercard via supported processors to ensure authorized transactions are processed and settled correctly.
- **Card lifecycle management:** COS support full card lifecycle management, enabling partners to control operations via COS APIs and web-based UI COS Explorer. **Webhook** integrations enable real-time event-driven updates for card lifecycle events, transaction activity, and status changes.

Partner's role

In the Integrated Issuing model, the partner's role is streamlined. Cross River manages the majority of operational, compliance, and technical infrastructure, including ledgering, transaction authorization, processor integration, and network settlement. This allows the partner to focus primarily on building the customer experience.

- **Customer experience and application development:** The partner is responsible for designing and maintaining the end-user interface, customer onboarding flows, and any custom logic for delivering card features within their application. Partners

integrate with COS APIs to create and manage card accounts, monitor status, and trigger lifecycle actions (e.g., replace, suspend) [1].

Find out more

- Card lifecycle
- Integrated card actions
- Integrated status and states
- Card transaction flow

Tutorials

See our tutorials to learn how to:

- Create a card
- Activate a card
- Simulate card management

Card lifecycle management

All card programs must support a complete lifecycle for each issued card, from creation to closure. The lifecycle governs the card's operational state. This card's operational state determines whether the card can be used for transactions, requires replacement, or must be suspended or closed due to fraud, customer request, or expiration.

In partner-managed programs, **lifecycle actions** are handled by the partner and their processor. The partner is responsible for triggering lifecycle events, maintaining card status, and communicating relevant updates to Cross River via daily reporting.

In integrated issuing programs, **lifecycle actions** are managed by the partner using Cross River's infrastructure, specifically, COS **APIs** or COS Explorer.

Typical lifecycle events include:

- **Card creation**: A physical or virtual card is generated and linked to a customer account.
  - Refer to the **create new card** API.
- **Card activation**: The card transitions from *Unactivated* to *Active* status and becomes eligible for transactions.
  - Refer to the **activate card by ID** API.
- **Replacement or reissue**: A new card is issued due to expiration, damage, or being reported lost or stolen.
  - Refer to **replace and reissue** for more details.
- **Suspension and unsuspension**: Temporarily disables or re-enables a card, often due to fraud or customer request.
  - Refer to the **suspend card** and **unsuspend a card** APIs.
- **Card closure**: Permanently deactivates a card. Closed cards cannot be reactivated.
  - Refer to the **close card** API.

Webhooks

Webhook integrations enable real-time event-driven updates for card lifecycle events, transaction activity, and status changes. You need to **register** to receive webhooks.

Find out more

- **Integrated status and states** - details on how lifecycle events affect transaction authorization.

3.2.1. Partner-managed

In Cross River's partner-managed model, partners are responsible for managing the card's lifecycle and communicating relevant updates to Cross River via daily reporting. There are a series of requirements that the partner must meet before a card can be issued to a customer.

Partners are required to ensure that:

- The customer profile is created.
- The customer passes the required OFAC/PEP screenings.
- The customer's card is associated with their COS customer record using the unique card token.

These steps are required to ensure Cross River can conduct oversight of the card program.

the customer is created, that they pass the required OFAC/PEP screenings and that a card associated to their COS customer record by way of providing a unique card token. These steps are critical to ensure Cross River can conduct transaction monitoring to meet our Compliance obligations.

Detailed requirements

	Requirement	Description
1	**Load customer data into COS**	Cross River performs global OFAC/PEP screens in our system (COS) at the time of onboarding and regularly after that.
2	**Add unique cardholder identifier into COS**	Cross River receives files from processors and from the networks with tokenized card numbers. The processor uses the tokenized card numbers as unique identifiers for each cardholder. To match cards with accounts, we need this unique identifier to be added to the cardholder's COS customer record. Cross River expects that whenever a new card token is generated, you update this COS customer record. It makes no difference if the card is new or reissued.
3	For Commercial Card Partners: **Link the business and cardholders in COS**	Cross River requires Commercial card partners to create both business and cardholder customer records in COS. Use the **beneficial owner API** call to update the COS customer record.

In the partner-managed model, when the processor creates the card in their system, we need the partner to create a customer in COS.

Card lifecycle and customer controls

Partners are responsible for defining and managing the full customer card experience through their processor [1], including:

- Card Issuance
- Card Activation
- Card Suspension and Unsuspension

- Administrative Blocks
- Card Closure and Replacement

> [1] This is subject to due diligence review and at in the sole discretion of Cross River

3.2.1.1. Load customer data

OFAC/PEP screening and loading customer data

To add customer data in COS:

1. After a successful KYC (know your customer) check, call the `POST /core/v1/cm/customers` endpoint in COS to create the cardholder's customer account.
2. OFAC uses the customer name, date of birth (DOB), and address. The Customer Identification Program (CIP) requires the customer social security number or government-issued ID (for non-US persons).
3. The webhook event `Core.Customer.Ofac.Changed` is triggered and returns an initial status of **Clear** or **Failed**.
4. If the status is **Clear**, you can approve and create the customer credit card facility on your side.
5. If the status is **Failed**, the Cross River AML team investigates the case. The webhook event is triggered again, sending a status of **Reviewing** or **Pending**, indicating that a person is reviewing the case. This can take up to 48 hours. On occassion, the Cross River team might reach out to you to ask for additional supporting documents. You will need to provide an email address for this. The `Core.Customer.Ofac.Changed` webhook returns any change in status, not just **Reviewing** or **Pending**.
6. After reviewing, the Cross River team moves the status to either **Clear** or **Positive**. The webhook event fires again.
   - If **Clear**, you can continue with approving and creating a credit card facility for your customer.
   - If **Positive**, you must decline the application. This is quite rare.
7. To trigger an OFAC/PEP failure in the Sandbox use these values in your call:
   - For PEP, in the `name` object use the name **Vladimir Putin** to get a hit.
   - For OFAC, in the `name` object use the name **Francisco Javier Arellano Félix**; and in the `profile` object use the birthdate **1969-12-11** (December 11, 1969)
   - For OFAC entity, in the `name` object use the `entityName` value **Blue Star Diamond**

8. Once the customer record is cleared, there is no longer a reason to subscribe to the specific customer webhook.
    - Cross River scans customer records nightly for OFAC/PEP. If a customer hits OFAC during a nightly scan, our AML team will reach out via Zendesk to help clear.
9. You must keep the customer records you create up to date.
    - If a customer changes their name, address, email, or other details, use the Customer Management APIs to update their record.
    - You can make this part of a nightly process.
10. The scans take less than a second. Anything that needs manual review is taken care of within 3 business days.
11. Status Definition Changes (applies to OFAC, PEP, and Internal List). Webhook event responses from this point on mean:
    - **Positive**: Must be rejected (no change)
    - **Clear**: Everything is fine (no change)
    - **Pending**: The case is being reviewed (no change)
    - **Reviewing**: A potential match within a category is being reviewed by CR and is pending a disposition. Previously this was Failed.
    - **Failed**: There isn't enough information to know the status (for example, RFI not responded to).
12. See the APIs required **to onboard a customer**.

As always, if you have any questions, email our **support** team.

3.2.1.2. Add ID details

Cross River receives files from processors and from the networks that contain tokenized card numbers identifying each cardholder. We need to add these unique user identifiers to the COS customer record of each cardholder. This allows Cross River to track all the information that the card networks need to know about that customer's card and how they use it.

Identifier

Add the identifier to the Customer ID section with this endpoint:

`POST /core/v1/cm/customers/{id}/identifications`

Use these values in the body of the request:

Curl

```
{
"isPrimary": false,
"idNumber": "<token value goes here>",
"idType": "Other",
"issuingAuthority": "TOKEN",
"issuingCountryCode": "US"
{
```

See our tutorial which explains how to **add ID details**.

As always, if you have any questions, email our **support** team.

3.2.1.3. Associate beneficial owner

When you provide cards to a business (and not directly to individual cardholders), both the business itself and each cardholder from that business have customer records in COS.

To ensure that the cardholder customer records are associated with the correct business customer record, you add each cardholder as a *beneficial owner* to the business record.

Associate the customer record

Associate the customer record with the business customer record using this endpoint from the **Add beneficial owner**

`POST /core/v1/cm/customers/{customerId}/beneficial-owners`

In the body of the request, use the customer ID of the business issuing the company card for the `customerId` value, and the customer ID of the cardholder for the `ownerCustomerId` value. Enter **Auth User** for the `ownerTitle` value.

Sample associate beneficial owner resource to customer record request

```
{
"ownerCustomerId": "a9da21a8-a9df-4db3-90f5-138d7cf4b10e",
"ownerTitle": "Auth User",
{
```

See our tutorial which explains how to **add a beneficial owner.**

As always, if you have any questions, email our **support** team.

3.2.2. Integrated card actions

Cards APIs

COS supports full card lifecycle management, enabling partners to control card operations via:

- COS APIs (direct system-to-system integration)
- COS Explorer (Cross River's operational UI)

Key lifecycle events include:

- **Card creation:** Physical or virtual.
- **Card activation:** Transition from Unactivated to Active.
- **Set card PIN:** Set intial PIN for the card.
- **Replacement or re-issue:** For lost, stolen, or expired cards
- **Suspension and unsuspension:** Temporary blocks
- **Card closure:** Permanent deactivation

> Once a card is closed, it cannot be reactivated. A new card must be issued if card access is needed again.

Card creation

For card creation using our **create a new card** API endpoint, you need a valid deposit account number, the configuration ID, and other card-specific information such as the name and address of the cardholder. A newly created card has the status `Unactivated`. When the card is successfully created, the order status is `OrderPending`.

> See our tutorial to learn how to **create a card**.

Card activation

Cards can only be used once they are activated. To activate the card, use the **Activate a card by ID** API endpoint with the COS card ID.

> See our tutorial to learn how to **activate a card**.

Set card PIN

Use the **Set initial PIN** API endpoint to set a customer personal identification number (PIN). This will be the customer's ATM PIN.

Card replacement

Order a replacement for damaged, lost or stolen cards using the **replace** API endpoint. Damaged cards remain active until the cardholder activates the replacement card. Lost or stolen cards are immediately closed and a replacement card is issued.

> **IMPORTANT**
> When you replace a card, you must make the replacement card the primary card in the Cross River system to avoid issues once the card is supplied.

3.2.3. Integrated status and states

Cross River defines several card statuses within COS that play a key role in how transactions are processed. These statuses help govern the card's lifecycle and support important compliance and security policies.

Each status directly impacts whether a transaction is approved and what message is sent back to the card network during a transaction attempt. COS uses this status system to make sure cards are managed securely and accurately.

Card status also helps determine the overall card state.

COS assigns both **statuses** and **states** to cards to govern their behavior during authorization:

State changes are communicated via webhook events (such as Cards.Status.Changed) to keep partner systems in sync.

Card status

The status refers to the card's activation level and whether the cardholder is allowed to use it. The card status also contributes to the **card state**.

The `status` attribute is the activation status of the card and if the card can be used by the cardholder:

- *Unactivated* - The card has not been activated by the cardholder
- *Active* - The card is active and ready for use
- *Suspended* - The card is currently suspended
- *Closed* - The card is closed

The `orderStatus` attribute is the current status of the order:

- *Order Pending* - The order is pending fulfillment by the processor

- *Completed* - The order has been fulfilled by the processor. For physical cards, the card is being printed and mailed to the cardholder
- *Failed* - The order failed at the processor level

The status of a newly created card is *Unactivated* and the order status is *OrderPending*. The order status updates to *Completed* when the order is complete. For physical cards, this means the card has been mailed to the customer.

Managing a card program may require you to update card statuses for specific scenarios. For example, you may want to offer your customers added security by allowing them to temporarily suspend a card that has been misplaced.

Update activated card status when necessary to suspend, block, or close a card:

- *Suspended* - Suspending a card temporarily inactivates it until you remove the suspension by activating the card again (unsuspending). The cardholder is allowed to remove this block.
  - Refer to the **suspend card** and **unsuspend a card** APIs.
- *Placed on administrative block* - An administrative block means that the card is both suspended and has been blocked using the `admin-block` endpoint. Only an admin can remove this block.
  - Refer to **admin-block** API.
- Closed
  - Refer to the **close card** API.

> **IMPORTANT**
> Once the card is closed, it cannot be reactivated.

Card states

A given card status can be due to more than one reason. For example, a card might be **suspended** because it was lost, stolen, or blocked due to fraud. Knowing the reason for a given status is important to manage the card life cycle. At Cross River we call combination of status and other attributes *card state*.

You see the card state in the `Cards.Status.Changed` webhook event.

> You need to **register** to receive webhooks.

Card state	Card status	Other related fields and their values
Card has never been Activated	Unactivated	
Card Activated	Active	
Card Closed	Closed	
Card Lost	Suspended	`StatusReasonCode` = **Lost**
Card Stolen	Suspended	`StatusReasonCode` = **Stolen**
Admin Blocked	Suspended	`AdminBlocked` = **true**
Fraud Blocked	Suspended	`FraudSuspect` = **true** `StatusReasonCode` = **FraudBlocked**
Card Suspended by User	Suspended	`StatusReasonCode` = **NotSet**
Card Unsuspended by User or Admin	Active	`StatusReasonCode` = **NotSet** `FraudSuspect` = **false** `AdminBlocked` = **false**

3.2.4. Replace and reissue

Order a replacement for damaged, lost or stolen cards using the **replace card by ID** `/cardmanagement/v1/cards/{id}/replace` endpoint. Damaged cards remain active until the cardholder activates the replacement card. Lost or stolen cards are immediately closed and a replacement card is issued.

> **IMPORTANT**
> When you replace a card, you must make the replacement card the primary card in the Cross River system to avoid issues once the card is supplied.

The following replace reasons are available:

- Damaged
- Lost
- FraudCompromised
- Returned
- Expired
- Stolen

3.3. Card transactions

A card transaction occurs when a cardholder uses their card to access stored value, deposited funds, or a line of credit. The general characteristics of transactions, such as the entities in volved in them and the main steps in the transaction lifecycle are described below.

Key players

- **Cardholder**: The person using the card. They present a card (physical or virtual), and their issuing bank checks if they have sufficient funds or credit to complete the transaction.
- **Merchant**: The entity accepting the card payment for goods or services. They work with an acquiring bank for participation in the card network ecosystem.
- **Card Networks**: Companies like Visa and Mastercard that route transaction data between banks. They set rules and security standards for card transactions.
- **Issuing Bank**: The bank that issued the card to the cardholder. It verifies funds or credit and processes the transaction and ensures card program compliance to relevant network and regulatory rules.
- **Acquiring Bank**: The merchant's bank that receives the transaction funds and deposits them into the merchant's account.

Main steps

A card transaction involves several steps to ensure the payment is processed securely and efficiently. Understanding these steps helps both businesses and customers ensure smooth and secure transactions. Here are the main steps:

1. **Authorization**: The cardholder initiates the transaction by presenting their card or providing their card credentials. The merchant sends these details to the payment processor, which checks with the issuing bank to verify if the cardholder has sufficient funds or credit available. If approved, a hold is placed on the funds or on the credit line.
2. **Clearing**: TThe transaction details are sent to the card network, which routes them to the issuing bank. The issuing bank confirms the transaction and prepares to transfer

the funds.

3. **Settlement**: The issuing bank transfers the funds to the acquiring bank via the card network (the merchant's bank). The acquiring bank then deposits the funds into the merchant's account.
4. **Record keeping:** Both the cardholder and the merchant receive records of the transaction. The cardholder sees the charge on their statement, and the merchant updates their sales records.

3.3.1. Card transaction flow

A card transaction flows through multiple systems to authorize and settle a payment. In the standard four-party model, such as that of the Visa or Mastercard networks, the participants include:

- **Cardholder** – the individual or business using the card.
- **Merchant** – the entity accepting the card.
- **Issuer** – the bank that issued the card (Cross River, via its BIN sponsorship).
- **Acquirer** – the merchant's bank or payment provider.

The process typically involves an issuer processor and an acquirer processor, which route the transaction through the card network (Visa, Mastercard) to complete the payment securely.

Authorization flow

The authorization phase determines whether a card transaction is approved or declined. This typically happens in milliseconds. Here are the steps:

1. The cardholder taps, swipes, or enters card info at the merchant's terminal or app.
2. The merchant sends transaction data to its acquirer processor.
3. The acquirer processor formats the request and sends it to the appropriate card network (Visa, Mastercard).
4. The card network forwards the request to the issuer processor associated with the BIN on the card for real time authorization.
5. The card issuer or issuing processor responds with an approval or decline decision to the card network.
6. The card network routes the decision back to the acquirer processor.
7. The merchant receives the response — and the transaction is approved or declined at the point of sale.

Transaction lifecycle

After a transaction is authorized, it may go through several additional phases depending on the program, network behavior, and any disputes:

- **Authorization:** The transaction is approved and a hold is placed on funds.
- **First presentment:** The merchant submits the transaction for clearing and settlement.
- **Clearing:** The acquirer sends the presentment file to the card network, which routes it to the issuer.
- **Settlement:** The issuer transfers funds to the network for any posted transactions while the network transfers funds to the acquirer.
- **Chargeback:** If the cardholder disputes the transaction, the issuer initiates a chargeback through the card network.
- **Representment:** The merchant can respond with evidence, re-presenting the transaction.
- **Arbitration (optional):** If the chargeback is unresolved, the network may arbitrate.
- **Adjustment (optional):** A final correction is made by the issuer or acquirer to resolve an error or reverse a prior action.

3.4. Reporting and compliance

As the issuing bank and BIN sponsor, Cross River provides access to the card network and regulatory oversight for card programs. Cross River maintains full regulatory and network compliance responsibility for all Partner-managed Issuing programs.

Partners must submit regular reporting to Cross River to maintain compliance with regulatory requirements, credit risk management obligations, and card network operating rules. Reporting ensures that Cross River can fulfill its role as issuing bank and BIN sponsor, maintaining oversight of partner program activity and risk exposure.

To meet these obligations, partners must submit reporting to Cross River on a regular basis. Required reports include:

- **Daily transaction and card balance files** for transaction monitoring and reconciliation
- **Monthly credit risk metrics** for credit card programs
- **Quarterly card network reports** for Visa and Mastercard compliance

Timely and accurate reporting is critical to maintaining a compliant card program under Cross River's sponsorship.

Daily reporting requirements

Partners (or their processors) must deliver two types of daily files:

- **Transaction master file:** Captures all balance-affecting transactions posted into the partner's or processor's ledger during the previous business day. This file supports transaction monitoring, fraud detection, and settlement processes.
- **Card balances file:** Provides the ending balance and status for each issued card at the end of each business day. This file supports account reconciliation, negative balance monitoring, and charge-off reporting, among other Cross River operational and compliance oversight functions.

Daily files must be securely delivered via Cross River's **SFTP** (Secure File Transfer Protocol) server, encrypted using PGP (Pretty Good Privacy), and formatted according to

Cross River's data exchange specifications.

Once a partner is set up for SFTP, the must ensure they have a PROD\CreditCardProcessing\DailyReports\ subfolder in their Cross River SFTP folder. This is where they will place the files. Partners should review the Cross River Card Files specification and data dictionary and submit sample files for validation to Cross River prior to launching their card program.

Primary uses for the Card Files include monitoring money movement and key data elements related to compliance and BSA/AML. It also allows Cross River and our partners to reconcile card program operation and account balances.

The Cross River Card Files must be formatted in a specific way. Broadly, Cross River prefers comma-separated value (.CSV) files delivered by the Partner or their Processor via Cross River's SFTP. Files should be PGP encrypted using Cross River Bank's public key. File naming convention is flexible, but we advise Partners to include the date in the name. Encrypted file names should be appended with .pgp. Lastly, the file names should be unique.

Cross River can work with our Partner to discuss adjustments to the files if their Processor already has files that fulfill the same purpose. Once the files are agreed upon, the partner and Cross River can kick off the project for data ingestion.

Monthly reporting requirements

Credit card programs only

> Monthly reporting is mandatory for credit card programs to maintain portfolio transparency and meet regulatory obligations.

For programs that issue credit cards, partners must also deliver a **Credit Risk Metrics File** on a monthly basis.

- **Purpose:** To provide Cross River with credit risk indicators that are not visible through daily transaction monitoring, supporting risk management and regulatory oversight.

- **Required data:** Metrics such as application volumes, approvals, declines, credit limits assigned, delinquency rates (30/60/90+ days past due), charge-offs, fraud losses, recoveries, and collection activity.
- **Format and delivery:**
  - CSV file format according to Cross River's specification.
  - Delivered to the /ClientCredit subfolder on Cross River's SFTP server.
  - All required metrics must be reported, with zeros for metrics not applicable.

Quarterly reporting requirements

All card programs

> All programs must submit quarterly network reports to ensure compliance with Visa and Mastercard operating rules.

- **Visa Quarterly Operating Certificate (QOC):**
  - Reports Visa-branded card activity associated with the BINs provided to the partner.
  - Must be submitted to Cross River no later than the 5th business day following the end of each calendar quarter.
  - File format: Visa-provided Excel template.
- **Mastercard Quarterly Mastercard Report (QMR):**
  - Reports Mastercard, Maestro, and Cirrus-branded card activity associated with the BINs provided to the partner.
  - Must be submitted to Cross River no later than the second Friday after each quarter ends.
  - File format: Mastercard-provided Excel template.

Quarterly files must be securely uploaded to Cross River's **SFTP** server, encrypted using PGP.

4. Payments

Cross River delivers a range of payment solutions to make transactions smoother. Our services cover both local and international payments, using different payment methods.

Instant payments

Cross River provides a domestic Instant Payments product. Individuals and businesses can transfer funds quickly between US-based financial institutions. These transactions clear and settle within seconds. Cross River works with two instant payment networks: RTP® via The Clearing House (TCH) and FedNow®. We provide network interoperability between the two, ensuring your instant payment is sent in the best way at all times.

Learn more

Card payments

P2C is the Cross River pay-to-card and pay-from-card solution for disbursement and collection of funds. P2C allows merchants to transfer money to and from debit card accounts. These transfers happen through participating card networks.

Learn more

International payments (cross border)

Cross River facilitates cross-border money movement using SWIFT and local in-country rails. Correspondent banking networks play a crucial role in these bank-to-bank settlements.

Learn more

ACH (Automated Clearing House)

ACH transactions involve moving money electronically between US bank accounts. People use ACH for direct deposits, bill payments, and recurring transactions. Using ACH through Cross River ensures compliance with ACH rules and regulations.

Learn more

Wires

Wire transfers are expedited electronic fund transfers between banks. They're commonly used for high-value transactions, international payments, and urgent transfers.

Learn more

XML batch payments

Cross River's ISO20022 payment channel provides you with a way to send all your instructions in one single file format (XML).

Learn more

Checks

Check writing and check deposits are features of the Cross River Account product. Together, these features allow customers to move money using a physical instrument known as a check.

Learn more

4.1. Instant payments

Instant payments APIs

Unlock the future of real-time payments by using Cross River's Instant Payments solution. You can instantly clear payments, 24/7/365 to ensure continuous availability. Cross River facilitates the transfer of funds in real time over RTP®, FedNow® and CRNow (book transfer) networks.

What Cross River offers

- Payment networks:
  - **RTP®:** High limits up to $1,000,000.
  - **FedNow®:** Broad reach, limits up to $500,000.
  - **CRNow (book transfers):** Internal transfers within Cross River, no inherent limits.
- **Network interoperability:** We offer you automatic selection of the payment network. which reduces development overhead and makes it easy to manage payments across all networks.
- **Send limits:** Each payment network has its own **transaction** send limits per transaction. Cross River sets a **daily** send limits for Instant Payments based on previous Instant Payments usage.
- **Payment queuing:** Instead of a payment being rejected if the participating bank is unavailable, Cross River **queues** the payment until the receiving institution is available. Cross River then automatically originate the payment. Queuing is available for both RTP and FedNow.
- **Fraud reporting:** Each instant payment network has specific requirements for reporting suspected fraud, with its own criteria for what qualifies as reportable fraud. Find out more from our **fraud reporting tutorial**.

Find out more

- **Send and receive**

- Payment types
- Payment networks

Tutorials

See our tutorials to learn how to:

- Send an instant payment
- Get service info
- Set payment expiration
- Fraud reporting

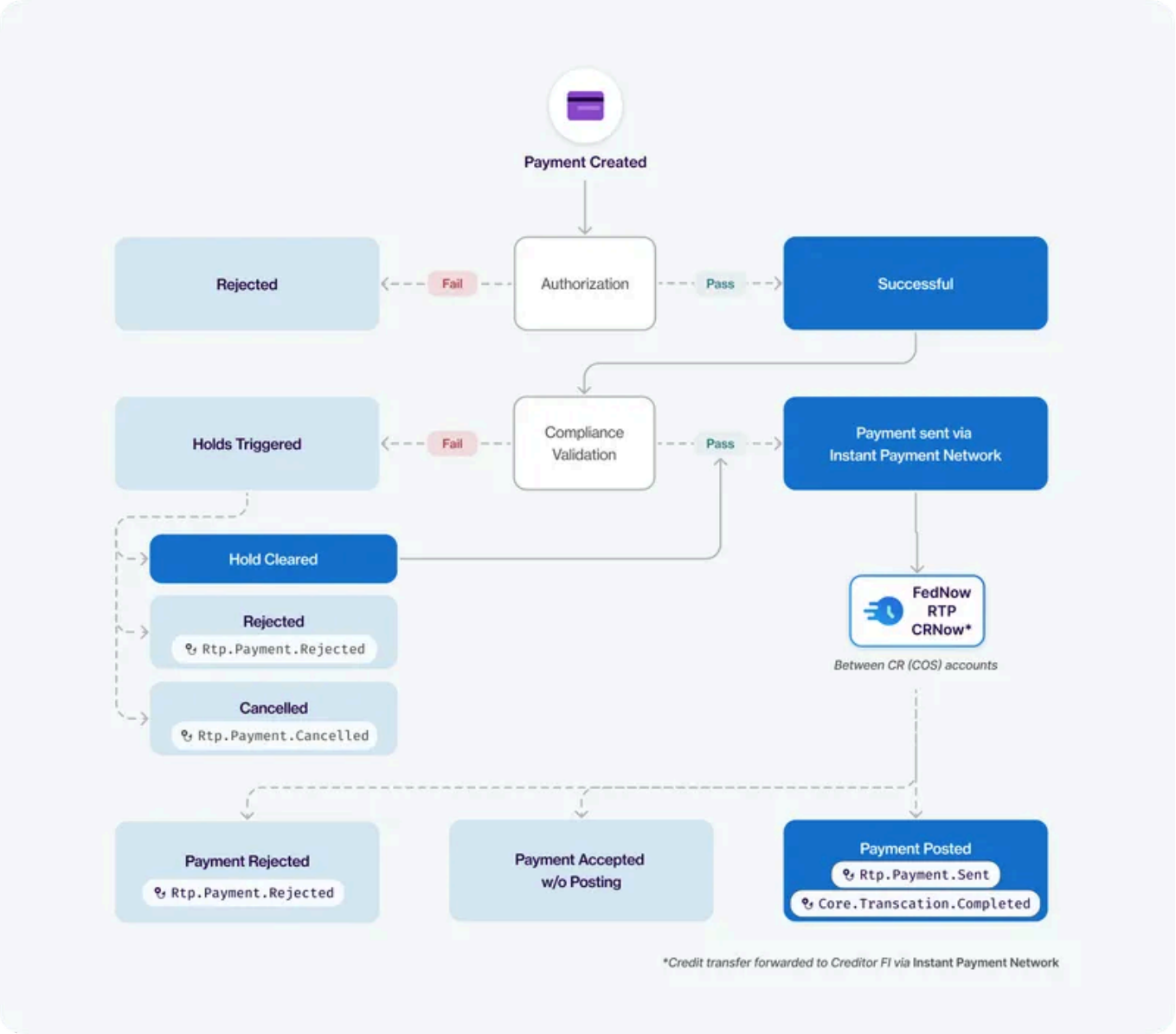
4.1.1. Send and receive

Cross River allows you to immediately send or receive funds via Book Transfer, the Federal Reserve's FedNow Service and The Clearing House's Real Time Payment Network. These transfers can be sent to and from Cross River 24/7/365 and the funds are credited instantly, as long as the CRNow, FedNow and RTP networks are operational and the requested service is **supported** by the receiving institution.

Outbound credit transfer

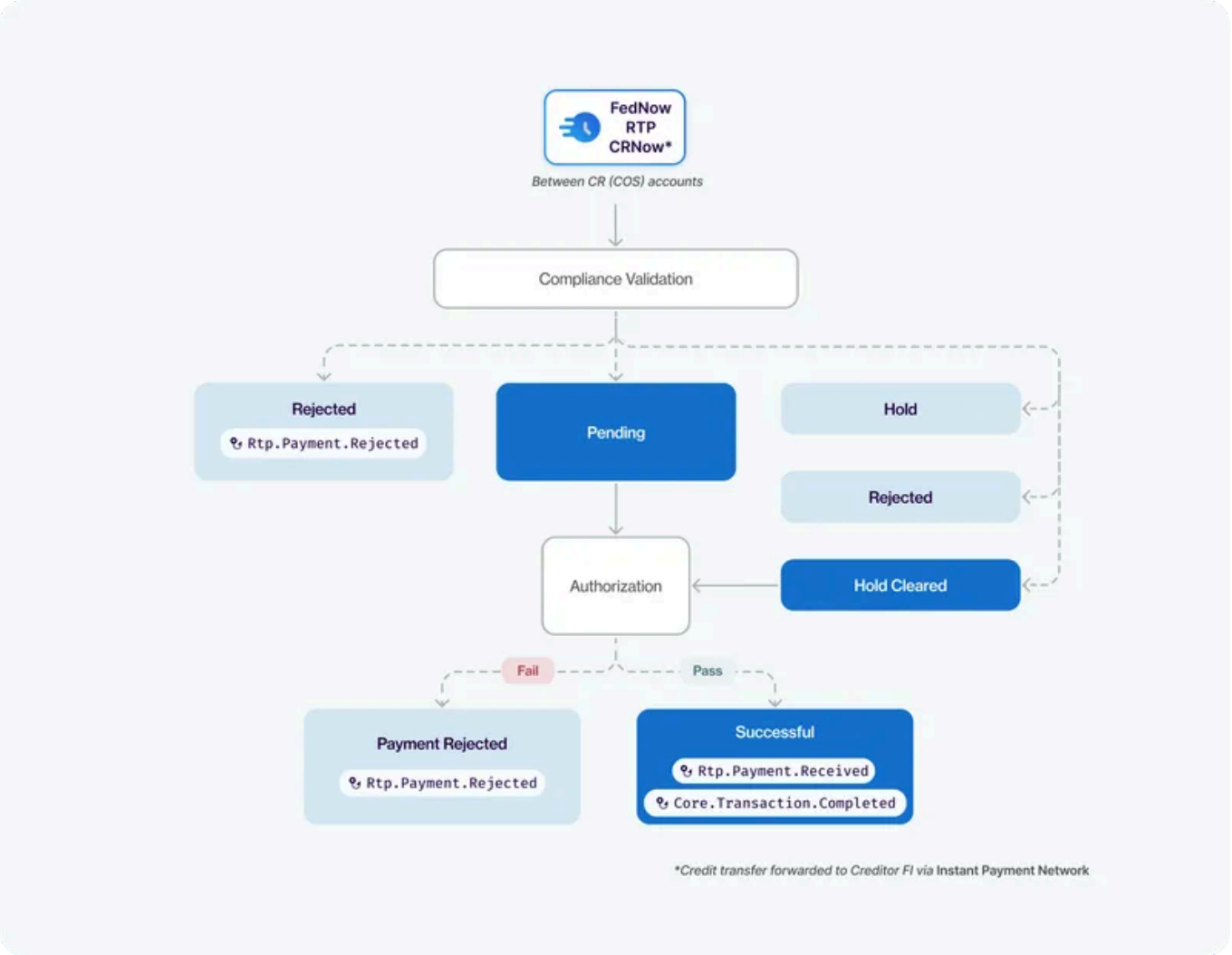
The following diagram shows the status and **webhook event** flows for outbound instant payment credit transfers.

> See our tutorial to learn how to **send an instant payment**.



Inbound credit transfer

The following diagram shows the status and webhook flows for inbound instant payment credit transfers:



FI supported services

Cross River provides a **directory API** to retrieve a list of TCH and FedNow services available from a financial institution. A successful API call returns a JSON response listing the supported services for the specific institution. If the institution participates in both networks, the response includes a separate list of services for each.

- The `receiveServices` object contains the service codes.
- The `networkPlatform` attribute specifies the Instant Payments network.

> Even if the financial institution lists a specific service, it does not guarantee that the creditor or debtor account at that institution is eligible for it.

Using our APIs you can see a complete list of **TCH services** and **FedNow services**.

4.1.2. Payment types

To send or receive instant payments you need to know the following basic information about credit transfers and **request for payments** (RfP) as well as the roles of the **ultimate debtor and creditor**.

Credit transfer

Instant Payments transfer funds in real time over networks like RTP®, FedNow®, or CRNow. This uses the **originate a credit transfer** API.

- **Message Type**: pacs.008
- **Purpose**: Sends funds from the *debtor* (account initiating payment) to the *creditor* (account receiving payment).
- **Debtor FI**: The financial institution sending the payment.
- **Creditor FI**: The financial institution receiving the payment.
- **Type of Payment**: Push payment (initiated by the sender).

Key notes

- Credit transfers are one-way payments initiated by the sender.
- Cross River enforces a daily send limit for Instant Payments per partner.
- CRNow facilitates instant credit transfers within Cross River's network, providing faster internal partner payments.

Refer to the diagram below for the end-to-end credit transfer process, showing interactions between the debtor, creditor, their financial institutions, and the instant payments network.

Request for payment (RfP)

Request for Payment (RfP) enables authorized pull payments by requesting funds from the debtor. This uses the **originate a payment request** API.

- **Message Type**: pain.013
- **Purpose**: Allows the creditor to send a payment request to the debtor, who must **authorize the payment**.
- **Debtor Role**: Reviews and approves the request before the transfer occurs.
- **Type of Payment**: Pre-authorized pull payment (initiated by the receiver through a request).

Key notes

- RfPs offer a way to collect payments securely and transparently.
- Instant Payments networks do not support automatic pull payments. Payments only proceed after debtor authorization.
- CRNow also supports RfPs for internal partner payments within Cross River's ecosystem, streamlining fund requests and transfers.

Refer to the diagram below that shows the flow from the initial request by the creditor to debtor authorization and final payment execution.

Ultimate debtor and creditor roles

Ultimate debtor and creditor roles provide receiving banks with greater transparency about the true originator or recipient of funds. These roles are essential when multiple parties are involved on either the sender (debtor) or receiver (creditor) side, clarifying which party is acting on behalf of another. This ensures precise and efficient transaction processing.

If the debtor or creditor account is a business account, then the ultimate debtor or creditor must also be a business account.

The ultimate debtor and ultimate creditor information is provided in the **instant payments** API.

Example flow

- The *Sender* (end user) initiates payment through the Merchant's app.
- The *Merchant* (Ultimate Debtor) facilitates the transaction on behalf of the Sender.
- *You* act as the Debtor, affiliated with Cross River.
- *Cross River* (Sending FI) manages and processes the transaction.
- The *Receiving FI* receives the funds on behalf of the final beneficiary.
- The *Receiver* (end beneficiary) collects the disbursement.

This diagram shows the fund flow variations involving Ultimate Debtors and Creditors.

4.1.3. Payment networks

Cross River gives you access to three different payment networks in a single APIs. RTP and FedNow are external networks. CRNow is an internal Cross River payment method that uses the same API as the instant payment networks.

Cross River uses the instant payment network capabilities to create and enhance digital services and give you the ability to send, clear, and settle payments immediately, 24/7/365.

RTP

The **real-time payments system** (RTP®) network via The Clearing House (TCH) is an instant payments platform. RTP includes advanced messaging capabilities and a focus on compliance.

FedNow

FedNow is the **instant payments network** operated by the Federal Reserve. Cross River was one of the first banks to join the FedNow network. The FedNow Service operates on good funds, meaning that funds must be available in the account sending the payment in order for the payment to be sent.

CRNow

Cross River ofter an internal payment network that enables partners to initiate instant funds transfers 24/7/365 without relying on external payment networks. CRNow leverages the our instant payment **originate a credit transfer** endpoint to support both inter-bank and intra-bank instant payments, providing partners with access to all instant payment services through a single interface. This simplifies integrations and enhances the overall customer experience.

4.1.4. Network interoperability

With Cross River's simple API, your company can use RTP®, FedNow®, or CRNow (book transfers) without having to worry about their interoperability.

Benefits of network interoperability

- **Efficient Routing:** Payments are routed based on network availability and product configuration.
- **Flexibility:** You can define routing preferences at both the **product** and **transaction** levels.
- **Reliability:** Payments leverage CRNow (book transfers), RTP®, and FedNow® to ensure seamless delivery.

By integrating advanced routing logic and supporting multiple networks, Cross River delivers a robust and flexible Instant Payments solution.

Routing preferences

Cross River offers two routing options for Instant Payments:

1. **Smart Routing:**
   Payments automatically route based on:
   - Configured network preferences
   - CRNow-specific criteria (for example, receiver account details, routing numbers, partner enablement)
2. **Explicit Routing:**
   You specify your preferred network for individual transactions by including the `networkPlatform` attribute in the specific API request.

Routing logic

All networks enabled

If you enable your product for all possible networks (CRNow (book transfers), RTP, and FedNow), routing is determined by:

- Receiver account details
- Routing numbers
- Partner configurations
- Availability of the creditor FI (or debtor FI for RfPs)

RTP and FedNow only

If you enable your product only for RTP and FedNow, routing follows these criteria:

- **In the API request:**
  - If the `networkPlatform` attribute contains a value in the API request, routing follows the value supplied.
  - If you don't provide the `networkPlatform` attribute, routing defaults to either your configured network preference for the product or Cross River's default.
  - See the **originate a credit transfer** endpoint.
- **Network selection when the payment is sent:**
  Routing follows these rules:
  - **Both networks available and online:** The transaction routes based on the configured network preference.
  - **One network available:** The transaction routes through the available network.
  - **No networks available:** The transaction fails.

4.1.5. Payment send limits

When you use instant payments, there are limits imposed both by the payment network and by Cross River.

Payment network transaction limits

Each payment network has its own limits for credit transfer amounts, per transaction.

Network	Limits
RTP (TCH)	• Transaction Limit: $1,000,000 per transaction. • Receiving Limit: Cross River adheres to this maximum limit for receiving funds.
FedNow	• Transaction Limit: $500,000 per transaction. • Receiving Limit: Cross River adheres to this maximum limit for receiving funds.
CRNow (book transfers)	• Transaction and Receiving Limits: No inherent limits, offering greater flexibility. • Partner-Specific Limits: Managed and periodically reviewed by Cross River's Credit Risk team.

Cross River daily send limits

Cross River sets a daily send limit for Instant Payments per partner to ensure smooth operations and prevent disruptions. Tracking utilization helps manage limits and prevent payment disruptions. We provide a **webhook** to allow you to monitor usage and to be aware when you are approaching your daily send limit.

- **Limit Determination** based on:
  - Previous instant payment usage.
  - Daily average balance in your Instant Payments-eligible account.

    - Projections provided during onboarding.
- **Adjustments:** Limits can be adjusted up or down with Cross River approval.

Cross River calculates your limit utilization by calculating the total amount of successful credit transfers and comparing it to your daily send limit for Instant Payments.

- **Calculation Frequency:** The system calculates utilization every 10 minutes for active products (products with successful transfers that day). This interval can be configured.
- **Processing Window:** Utilization resets at **1 pm (ET)** each day. Payments are tracked based on a timestamp.
- **Data Retention:** Utilization history is stored for 28 hours. Data older than 28 hours is removed hourly.

4.1.6. Payment queueing

Participating banks must be online to send and receive instant payments. Occasionally, a bank may be offline due to maintenance or technical issues. When this happens, Cross River queues instant payments until the bank is back online. This way you don't have to worry about resending a payment if the receiving bank (RDFI) is temporarily unavailable.

How it works

Instant payment queuing follows a structured process based on timing and the receiving bank availability:

1. You submit a payment to Cross River.
   When you submit the payment, you can also set a queue expiration time to cancel the payment if the receiving bank remains offline.
2. If the receiving bank is offline, the payment is queued and you get a notification.
3. If the receiving bank is online, the payment proceeds immediately without being queued.

Cancel a payment

You can cancel a queued payment before it reaches its expiration.

How long can a payment stay queued

If you don't set an expiration time, payments have a default queue expiration of **3 days**, after which the payment is cancelled.

Set a custom queue expiration

You can set one or both of the following:

- A specific date and time

- An expiration time in seconds

If you provide both, the system prioritizes the **time in seconds** and ignores the date and time.

Set the custom queue expiration using our **originate a credit transfer** API.

4.2. Card payments

Card payments APIs

Cross River has both a pay-to-card and pay-from-card solution for disbursement and collection of funds. The solution is sometimes referred to as our P2C solution. P2C allows merchants to transfer money to and from debit card accounts. These transfers happen through participating card networks.

What Cross River offers

Payment types

- **Push transactions:** When you send funds directly to a debit card, we call this a **push** payment or a push-to-card (P2C) transaction.
- **Pull transactions:** are also known as an Account Funding Transaction (AFT). It lets you **pull** funds from a debit card for specific purposes.

Payment options

- **International transactions**: Also known as cross-border payments, these are transactions in which the payer and the payee are based in different countries or are using different currencies.
- **Partial payment authorization**: Cross River partners with credit card networks to offer *partial authorization* of AFT (pull) transactions. Should a user account have less than the transaction amount available, participating platforms authorize their approved users to transfer a part of the transaction amount instead of the full sum. The user pays the rest using a different form of payment.
- **Flexible attributes for partners with submerchants:** Send different **business types** (BAI codes), **statement descriptors**, CAIDs, or MCC codes all in a single program as approved by the card networks.

Payment security

- **Fraud and risk detection**: Cross River designed its fraud detection services to protect online card payments. Our aim is to provide our partners with peace of mind while they conduct digital transactions.
- **iFrames**: An iFrame makes embedding payment functionality into websites convenient and straightforward. Through secure handling of payment information, iFrames ensure PCI compliance. Card details that a customer enters go directly into the iFrame.
- **Tokenization**: To avoid fraud, Cross River allows the use of tokens to secure debit card numbers. Stored by the merchant, a token identifies a card in place of an actual card number. Our in-house tokenization to securely access payment credentials without storing PANs (Primary Account Numbers).
- **Card authorization and compliance:** P2C processing complies with Payment Card Industry (PCI) standards to ensure that sensitive credit card information is protected from fraudulent use. Cross River uses only encrypted token information throughout our system. Any card number we use to transfer funds is meant for temporary access only during a transaction request.

Find out more

- **International transactions**

Tutorials

See our tutorials to learn how to:

- **Send a push transaction**
- **Send a pull transaction**
- **Set up iFrame**

4.2.1. Send and receive

Push payments

When you send funds directly to a debit card, we call this a push payment, or a push-to-card (P2C) transaction using the **send funds to a payee** API. The issuing bank (of the registered debit card) receives the transaction. The issuing bank either authorizes or declines the transaction. If the bank authorizes the transaction, your payee will receive the funds. A push payment is also known as Original Credit Transaction (OCT).

> Original credit transactions (OCTs) reversals can occur only under specific circumstances, and at the discretion of the card issuer. Once the funds have been delivered to the cardholder, OCTs cannot be reversed through the Card Payments API. For assistance in requesting a reversal on a specific transaction, please contact your Cross River relationship manager.
>
> Download Cross River's OCT fraud submission protocol:
>
> CrossRiverOCTFraudSubmissionProtocol...

Pull payments

Are also known as an Account Funding Transaction (AFT). It lets you pull funds from a debit card for specific purposes. You can use an AFT to fund a wallet or a prepaid card through the **request funds for a payer** API . Or, you can use it to initiate a person to person (P2P) transfer through the card network rails. AFTs are not allowed for buying goods and services.

4.2.2. International transactions

An international transaction involves one or both of the following:

- *Multi-country*. A cross-border transaction, where the country of the card issuer determines the conditions of the transaction.
- *Multi-currency*. Transactions in currencies that the payments processor supports.

Functions

The following functions, which the **card payments API** supports, are part of international transactions.

- **OFAC**. Check whether a person or organization in an international transaction is on a restricted list. OFAC is part of the US Treasury Department. OFAC keeps this list for security reasons and to prevent financial crimes.
  For cross-border payments, OFAC is *mandatory*.
- **Foreign exchange rates**: Access the current conversion rate for a pair of currencies. The FX Rates API returns foreign exchange rates, markup fee, and disclosure data. It enables you, the merchant partner, to fully disclose to the customer the rates they should expect to pay.
  For cross-border payments, Foreign exchange (FX) Rates in *recommended*.

> The same Cross River Transaction APIs serve domestic as well as international transactions.

Currency exchange

Sometimes, a transaction happens with a US-issued payment card outside the US. Or, it can occur within the US using a non-US-issued card. These transactions require changing from one currency to another.

The foreign exchange (FX) rates API makes it easy to access the conversion rate for the currencies of two different countries.

Currency conversion rates are indicative. This means they are dynamic, not fixed.

Conversion rates depend on:

- The currency pair
- The network exchange rate

Real-time exchange rates

Cross River provides you with the real-time foreign exchange rate in the currency of the party receiving the payment. Cross River FX Rates reflect the rates now available in financial networks.

In the case of Visa and MasterCard, the API responds to a function call with a list of rate data from the card networks. These rates can differ between networks. They are updated once each day. The Forex Rates API provides clear information about any network processing transactions.

> **Mastercard**. Rates change every day at **14:05 CST** in the US. They remain valid for **24 hours**. Let's say a transaction occurred on May 1st, 2023 at 3:00 AM CST. The rate that applies to the transaction would be the rate set on April 30th, 2023. That rate covers the period from 14:05 CST on April 30th to 14:05 CST on May 1st.
>
> **Visa**. Rates change once a day at around **19:00 EST** (midnight GMT). Over the weekend, the system uses the rates from Friday.

Use case

For example, suppose a customer from Sweden wants to buy products from your American online store. As a merchant partner, you need to perform an international transaction. That's because the customer is buying in kroner (the currency of Sweden) products being sold in US dollars.

Cross River provides a call to the FX Rate API endpoint. You make this call before performing the transaction request.

These rates are most useful when working with multi-currency or cross-border transactions.

> In addition to the current exchange rate, the FX Rates API also reflects the markup fee a network charges for a foreign transaction.
>
> For transaction settlement, the currency conversion rate returned by the API may differ from the actual rate of exchange offered in the market.
>
> The FX Rates function call is standalone. It returns the exchange rate without actually determining the international transaction.

API reference

Learn about the **foreign exchange rates** API endpoint.

4.2.3. OFAC screening

OFAC APIs

The Office of Foreign Assets Control (OFAC) is a financial intelligence and enforcement agency of the US. Treasury Department. OFAC compiles a list of individuals and organizations who may present security and criminal risk to society. They do this to enforce economic and trade sanctions in the United States. Those entities are forbidden to make financial transactions in the US. The OFAC list enables financial institutions to monitor and restrict the use of their services to entities on the list. Individuals or organizations who are on the OFAC list may include terrorists, politically exposed foreign officials among other criminals.

OFAC API function calls

Cross River enables and supports **OFAC API calls** to comply with sanction screening obligations. These are the function calls that let you verify OFAC compliance, get screening status, and get scan ID. You need to meet these requirements when executing monetary transfers that include non-US countries.

Use of these calls is important! Both the US and Canada administer and enforce economic and trade sanctions. They base these sanctions on foreign policy and national security goals. As such, both countries take measures against a range of targets that include:

- Foreign countries and regimes
- Terrorists
- International narcotics traffickers
- Those engaged in activities related to the proliferation of weapons of mass destruction
- Other threats to each country's national security, foreign policy, or economy

It is strictly prohibited to do business with countries or entities on an OFAC list. Likewise, it is forbidden to facilitate financial transactions to or from individuals, entities, or countries on these lists.

OFAC violations can result in severe civil and criminal penalties!

Call the OFAC endpoint

If you want to do international transactions, you must execute a function call to the **OFAC endpoint** *before* the transaction request.

The OFAC screening endpoints perform OFAC screening to a specific cardholder. The endpoints respond with that person's OFAC status.

The card payments gateway validates the international transaction recipient-sender pair with recent OFAC calls. That is how the gateway verifies the validity of the pair. Let’s say the cardholder was not scanned recently for a transaction or was scanned but was not validated. In that case, the gateway will reject the international transaction, providing the reason for the rejection.

4.2.4. Partial payment authorization

Cross River partners with credit card networks to offer *partial authorization* of AFT (pull) transactions. Should a user account have less than the transaction amount available, participating platforms authorize their approved users to transfer a part of the transaction amount instead of the full sum. The user pays the rest using a different form of payment.

Benefits to you

Reduced NSF declines: Instead of having to decline and thereby lose the transaction, your users can still make the pull funds transaction if they have an additional form of payment.

Better consumer experience: Consumers are less embarrassed than when their card is declined. More comfortable consumers means more revenue.

Improved authorization performance: An increased approval rate improves your merchant portfolio scoring.

> Currently only Visa offers partial authorizations

Key advantages

- Provides an alternative to a declined transaction when the available Visa card or account balance is not sufficient to approve a transaction in full.
- Improves AFT authorization performance by reducing the declines due to insufficient funds.
- This value added service comes at no cost to you.

Relevant API endpoint

To request partial transfer of funds, use the **request funds from a payer** API for a pull transaction and include the optional field: `acceptPartialAmount`. This field is only sent to participating issuers.

4.2.5. Business types

Card payments APIs

Business types define the purpose of transactions according to specific categories. The business type is passed on to the card network when making a transaction. Business types are sometimes called *use cases*.

In our APIs, the `businessType` identifier is a two-character code Cross River uses to identify the purpose of a push or pull payment. We ultimately pass these to the Visa Direct and MasterCard Send payment rails as BAI codes (Visa) or TTIs (MasterCard).

What the business type code does

Business type codes determine the following:

- The data that a push or pull payment message must contain.
  Some business types require additional information in the request. For example a value of **PP** for the `businessType` means the `correspondingEntity` object is required.
- The limits and economic conditions relevant to a particular transaction.

Based on the business type, a payment sender or receiver can make an authorization decision.

Who provides the business type

The possible business type for a transaction is determined together with Cross River and the card networks during onboarding and is based on the use case of the money movement.

If you are approved for only a single business type, Cross River configures this for you, so the information properly passes to the card networks. This is your *Default Business Type*. In this case it is not necessary to pass the `businessType` in the API request.

If you are approved for multiple business types, you can pass the `businessType` code in the API request, to be passed along to the card networks.

Business type codes

Code	Application type	Payment category
AA	Account to account	Money transfer
BB	Business to business	Funds disbursement
CD	Cash deposit	Money transfer
CP	Card bill payment	Money transfer
FD	Funds disbursement (general)	Funds disbursement
FT	Funds transfer	Money transfer
PP	Person to person	Money transfer
WT	Wallet transfer	Money transfer
BI	Money transfer (bank-initiated)	Money transfer
CI	Cash In	TDB
CO	Cash Out	TDB
GD	Government disbursement	TDB
GP	Gambling payout	TDB
LO	Loyalty credits and rebates	TDB
MD	Merchant disbursement	TDB
MP	Face-to-face merchant payment	TDB
OG	Online gambling payout	TDB
PD	Payroll Disbursement	TDB
TU	Prepaid Card Load or Top-Up	Money transfer

In the framework of *money transfers*:

- AA applies to transactions where sender and recipient are the same person.
- PP is when sender and recipient are *not* the same.

Corresponding entity

In card payments, corresponding entities are either the receiver of a pull transaction (AFT) or the sender of a push transaction (OCT). This information must be included in the request for money transfers (business type code: AA, PP, BI, CD, FT, WT, CP, TU) in the `correspondingEntity` object of the request.

4.2.6. Statement descriptor

A *statement descriptor* is the text that appears for each transaction listed on the monthly statement sent to your customer. The statement descriptor gives the cardholder information to identify each transaction.

You can configure the text of a statement descriptor for your organization.

Statement descriptions are important because:

- Bank customers need an easy way to identify the charges that appear on their bank statements.
- Banks and card networks need information to help end-customers understand their bank statements.
- Chargeback and return-risk increases if end-customers don't recognize a transaction by its description.

Cross River supports 2 kinds of statement descriptors: *static* and *dynamic*.

When you onboard to Cross River you supply your merchant name and abbreviated name. Tell your payments implementation manager whether you prefer a static or dynamic descriptor. If you're configured for dynamic descriptors you enter your additional text as part of the API request for each transaction.

Static descriptor format

This text is always the same and also appears as part of the dynamic descriptor. Tell your implementation manager which types of static descriptor you prefer:

- Merchant name
- Abbreviated name and merchant name

For example, if the merchant name is **Cross River** and the abbreviated name is **CRBT**, the descriptor could appear in these ways:

- `CrossRiver*`
- `CRBT*CrossRiver`

Some processors automatically append city and state to the descriptor:

- `CrossRiver* Newark, NJ`
- `CRBT*CrossRiver Newark, NJ`

Dynamic descriptor format

If you're configured for **dynamic statement descriptors**, you first choose whether to include the abbreviated name or not. Like with static descriptors, this text always appears for every transaction and does not change. In effect, the dynamic descriptor is in addition to the static descriptor.

4.2.7. Fraud and risk detection

Welcome to Cross River fraud detection services (FDS). Digital transactions are always getting more complex. This exposes merchant partners—our customers—to greater risk. Ensuring the security and trust of our customers and their cardholders is always Cross River's top priority. Therefore, Cross River designed its fraud detection services to protect online card payments. Our aim is to provide our partners with peace of mind while they conduct digital transactions.

Key services

These are the key services making up the Cross River fraud prevention strategy:

- **Token validation service (TVS)**.
  Tokenization is a powerful security measure. Tokenization replaces sensitive card data with unique security tokens. Cross River TVS ensures that the token used in a transaction is authentic and hasn't been tampered with. This reduces the risk of data breaches. That's because the actual card details remain stored in way that's secure.
- **OFAC (Office of Foreign Assets Control) compliance**.
  OFAC makes sure that card payment transactions comply with the law. Cross River OFAC screening APIs check transactions against the OFAC list. That list contains sanctioned people and organizations, meaning, those not allowed to make transactions in the United States. OFAC screening is proactive. It identifies and prevents financial interactions with restricted entities. Through OFAC, Cross River maintains the integrity of transactions taking place in its banking ecosystem.
- **Address verification service (AVS)**.
  AVS plays a key role in stopping fraudulent card payments from happening. During a transaction, the consumer provides a billing address. They do this through the merchant's website. AVS checks that address's accuracy against the address that's in the hands of the card issuer. By confirming this information across systems, Cross River can detect mismatches instantly. Those differences could indicate a possible unauthorized transaction.
- **Account name inquiry (ANI) service**.
  ANI confirms that the account holder's name is the same one listed at the issuing bank. This provides an added layer of checking. This check is important because it

helps identify mismatches between the cardholder's name and the transaction details. Here too, those differences could indicate a fraud attempt.

- **Card validation value 2 (CVV2)**.
  CVV2 offers extra defense against unapproved card usage. The CVV2 is a three- (or four-) digit code. That code appears on the back of a payment card. When buying something with their payment card, the consumer has to enter this code into the merchant's website. As the merchant acquiring bank, Cross River forwards this code to the card network. Then, the network can get payment approval from the cardholder's bank. CVV2 ensures that the person making the transaction physically has the card. As a result, no one needs to rely on card information that might have been stolen.

Address verification (AVS)

During card payment transactions, Address Verification Service (AVS) checks the validity of the billing address that the cardholder provided. AVS plays a key role in Cross River's efforts to prevent fraud and ensure transaction security. This simple yet powerful tool strengthens security, helping keep transactions safe. As a fraud detection measure, AVS helps make Cross River's entire ecosystem a safe environment for its merchant partners.

Importance of AVS

When customers transact online, the card is not visible to the merchant. By confirming card details with the issuing bank online, we can detect possible fraud activity. We can also find out if it's likely the transaction is unauthorized. AVS helps protect consumers, our merchant partners, and Cross River.

How AVS works

When a customer enters their billing address, Cross River sends it to the card issuing bank. The issuing bank, in turn, compare the address with the one they have on file. If the address is a match, the transaction proceeds smoothly. However, if there is a mismatch, this could be a sign of potential fraud. Note that non-matching AVS responses will not decline a transaction automatically. That's because there could be a few valid reasons for the mismatch. For example, the consumer may have mixed up digits in the zip code.

Ultimately, it is up to the merchant to decide whether to accept the transaction. They make this decision based on AVS results as well as other key data.

The results of checking AVS are held in the Add a card (POST API) function call. This call contains the following key attribute as part of its response:

`AddressVerfied` (true/false). If the merchant isn't configured for the AVS service, contact your Cross River relationship manager for details on how to connect to AVS.

Key benefits of AVS

- **Prevent fraud**. Catch suspicious activities early, preventing fraudulent transactions from occurring.
- **Protect customers**. Ensure that transactions are authorized, protecting consumers from unauthorized charges.
- **Save costs**. Detect fraud before it happens. This saves money and prevents losses for everyone involved in the card payments process.
- **Ensure trust**. Consumers trust their merchant sites a lot more when they know their acquiring bank is working to keep transactions secure.
- **Promote efficiency**. Automate the verification process. This makes transactions go more smoothly for legitimate consumers.

Account name inquiry (ANI)

The ANI service matches and verifies the name of an account holder. It checks whether the name on file at the issuing bank is the same as the one the consumer provides during a transaction. This happens in a way that is similar to address verification (AVS). ANI is vital for ensuring secure and accurate card payments.

Importance of ANI

Similar to AVS, ANI strengthens security, minimizes fraud risk, and maintains the accuracy of transactions. This helps create a safe payment experience that consumers, merchant partners, and card issuers can trust.

ANI also helps check for ineligible transactions. For example, a cardholder requesting earned wage access via push-to-card may have given an incorrect name. This could indicate that the cardholder-provided card details do not really belong to the cardholder.

How ANI works

Cross River always initiates an ANI verification transaction during card authorization. This step is essential before making any AFT or OCT request. This compares if the names that the transaction initiator provided match the data held by the card issuing bank.

If a card issuer doesn't yet support the ANI service, the merchant gets a response code. This code indicates that the cardholder's name is unverified because their card issuing bank doesn't support ANI.

The results of checking ANI are held in the Add a card (POST API) function call. This call contains the following key attribute as part of its response:

- `nameVerificationUsed` indicates whether Cross River tried to contact the ANI service. This attribute value will also let you know if the merchant isn't configured for the ANI service. In that case, we recommend contacting your Cross River relationship manager for details on how to connect to ANI.
- `nameVerificationResult` indicates whether the initiator-provided card details match fully, partially, or not at all, or if they are unknown.

Key benefits of ANI

- **Enhance security**. Adds extra security by verifying the account holder's identity.
- **Prevent fraud**. Detects unauthorized transactions in real-time, preventing the loss of money.
- **Build customer trust**. Verifies account details before problems occur.
- **Ensure accuracy**. Reduces errors in transactions by ensuring the person making the transaction is the real account holder.
- **Comply with standards**. Meets security and verification standards for payment transactions, such as KYC, AML, POA, and so forth.

Card validation value (CVV2)

The Card Validation Value 2 (CVV2) service provides a critical security layer in card payments. Cross River provides CVV2 validation to ensure safer credit and debit card transactions. CVV2 is a unique 3-digit (or in the case of Amex 4-digit) code printed on the back of all cards. During online transactions where the card isn't physically present, CVV2 helps authenticate the cardholder. It does this by making sure the cardholder has the card with them at the time they are initiating the transaction. This deters fraud. It also increases cardholder confidence when consumers make online purchases or card payments.

Importance of CVV2

CVV2 confirms that the person using a card physically has the card. The CVV2 appears only on the back of the actual card and in the issuing bank's files. It doesn't exist anywhere else. This way, requiring the CVV2 code adds an extra level of protection against unauthorized transactions initiated from stolen card credentials.

How CVV2 works?

When a consumer makes an online or phone transaction, the merchant requires them to provide the CVV2 code. The merchant provides this code as part of the transaction data sent to Cross River. The card issuing bank, in turn, checks that the CVV2 code the consumer enters matches the code on-record. Then, the issuing bank returns a match or mismatch response. Finally, Cross River provides this matching response to the merchant.

Note that card issuers may still approve transactions with mismatched CVV2 codes. It is the merchant's responsibility to consider the matching data when deciding if to proceed with the transaction.

Key benefits of CVV2

- **Stronger security**. CVV2 protects transactions by requiring the unique code for verification.
- **Deterring fraud**. CVV2 discourages unauthorized users from trying to make transactions that are not allowed.

- **Card-not-present protection**. CVV2 isn’t stored electronically or magnetically. This way, it protects online and phone transactions. It does this by requiring a code that only the cardholder should know.
- **Consumer confidence**. Consumers feel more secure using their cards knowing CVV2 is available and being enforced by merchants and banks.
- **Reduced fraud costs**. Preventing fraud prevents loss of money for everyone involved in the payment chain. This holds true for the consumer, merchant, and bank.

4.2.8. iFrames

An iFrame makes embedding payment functionality into websites convenient and straightforward. Through secure handling of payment information, iFrames ensure PCI compliance. Card details that a customer enters go directly into the iFrame. Cross River sends merchants a token that represents those details. Then, merchants can use that token to request a card-based transaction.

> **IMPORTANT**
> In Cross River's Card Payments API, iFrames are compliant with the PCI DSS standard. PCI-compliant iFrames lower the cost of PCI compliance by merchants. That's because credentials are already saved onto a Cross River server. This means those credentials don't have to be stored by CR partners.

As a Cross River merchant partner, use iFrames:

- If you are not PCI-compliant
- For quick and easy payment form customization
- For another way to reach your customers, in parallel with your existing online registration.

iFrame process

1. Host an iFrame on your website.
2. The cardholder—the customer—enters their card details directly into the iFrame. You don't store, process, or send the customer's card data on your own form or server.
3. Cross River processes customer-sensitive information and returns a token representing those details. Cross River does this through a process called *tokenization*.
4. A webhook sends you a card token.
5. Use the card token to make a payment request.

> See our tutorial to learn how to **generate the iFrame** and the related **APIs**.

iFrame enabling

When you enable an iFrame, a cardholder signs up their card to your website. As a result, card details are entered instantly to your system for secure processing.

Every time a cardholder enables an iFrame session, Cross River assigns a one-time code for that session. Then, the code becomes associated with the payee's card.

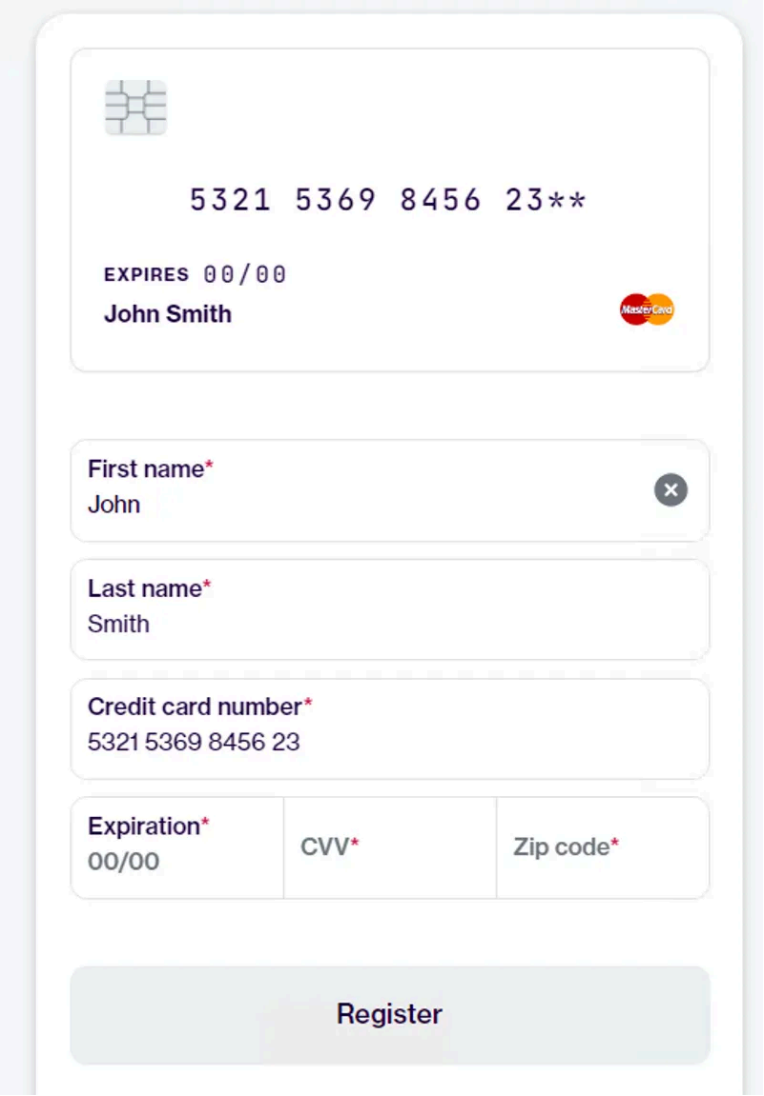
When the cardholder's browser encounters an iFrame element, it creates a new HTML document environment. The content that the iFrame contains gets loaded into that document.

iFrames key benefits

- iFrames integrate smoothly with your existing site.
- iFrames are an open technology that all major web browsers support.
- With the Cross River iFrame Generator:
    - Once you create an iFrame template, you always control the look and feel.
    - Need to make changes? Do it via the **iFrame Generator UI**.
    - You never have to touch the code!

Use of iFrames

Follow the configuration steps (described in **Generating an iFrame** below). After that, integrate the iFrame into the look and feel of your website.



Sample iFrame image

Generating an iFrame

To set up an iFrame and get it working on your merchant site, use the iFrame Generator and **generate an iFrame OTC signup** API.

iFrame Generator—Introduction

Overview	**Welcome to the iFrame Generator, your self-service interface for creating PCI-compliant iFrames. iFrame Generator provides an efficient and streamlined way to embed iFrames onto your merchant website.** **With iFrame Generator, creating an iFrame is a one-time action. Once you have generated an iFrame, you can immediately begin entering customer card information via your website. Of course, if you wish, you may add more iFrames in the future.**
Use case (high-level)	Do the following to create and embed an iFrame onto your site: 1. Create an iFrame design template. 2. Customize the template based on the requirements of your merchant website. 3. Save the template. 4. Download the iFrame code. 5. Embed the iFrame code into your website.

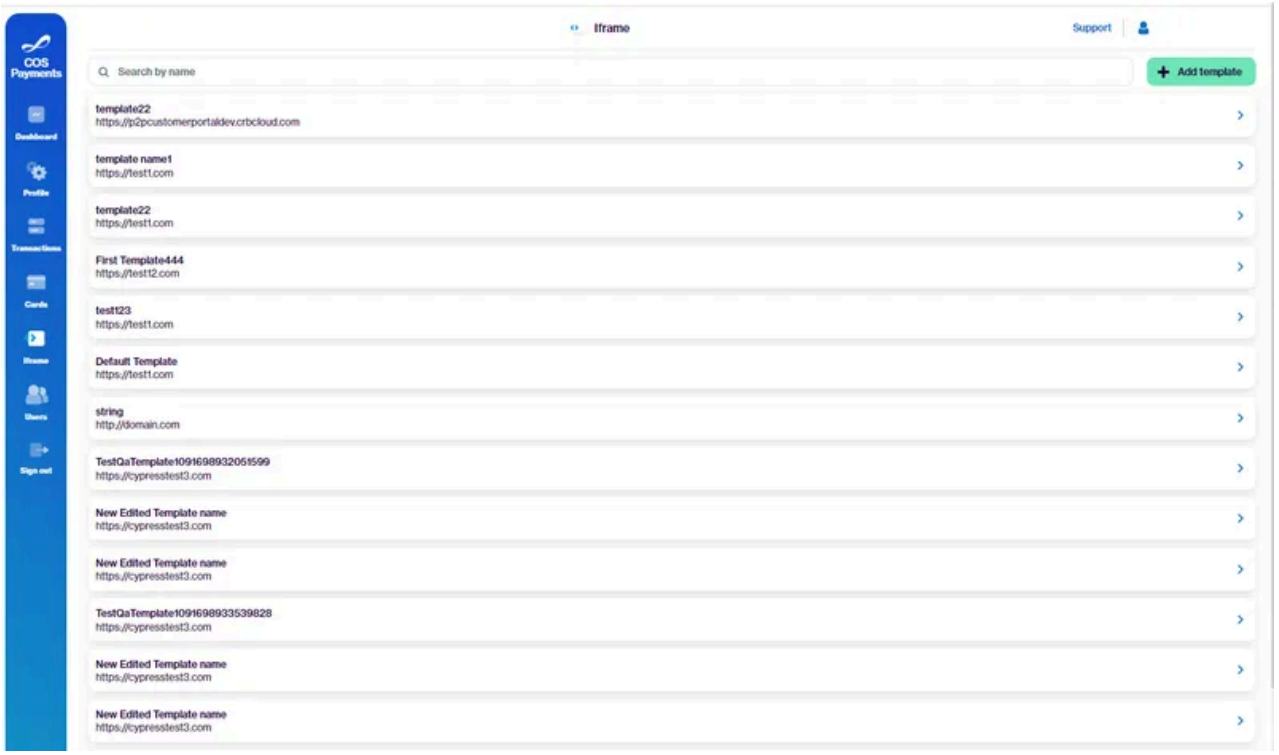
iFrame Generator—Procedure

The procedure for generating an iFrame contains 3 main steps:

1. Add a template.
2. Define a style.
3. Build a URL.

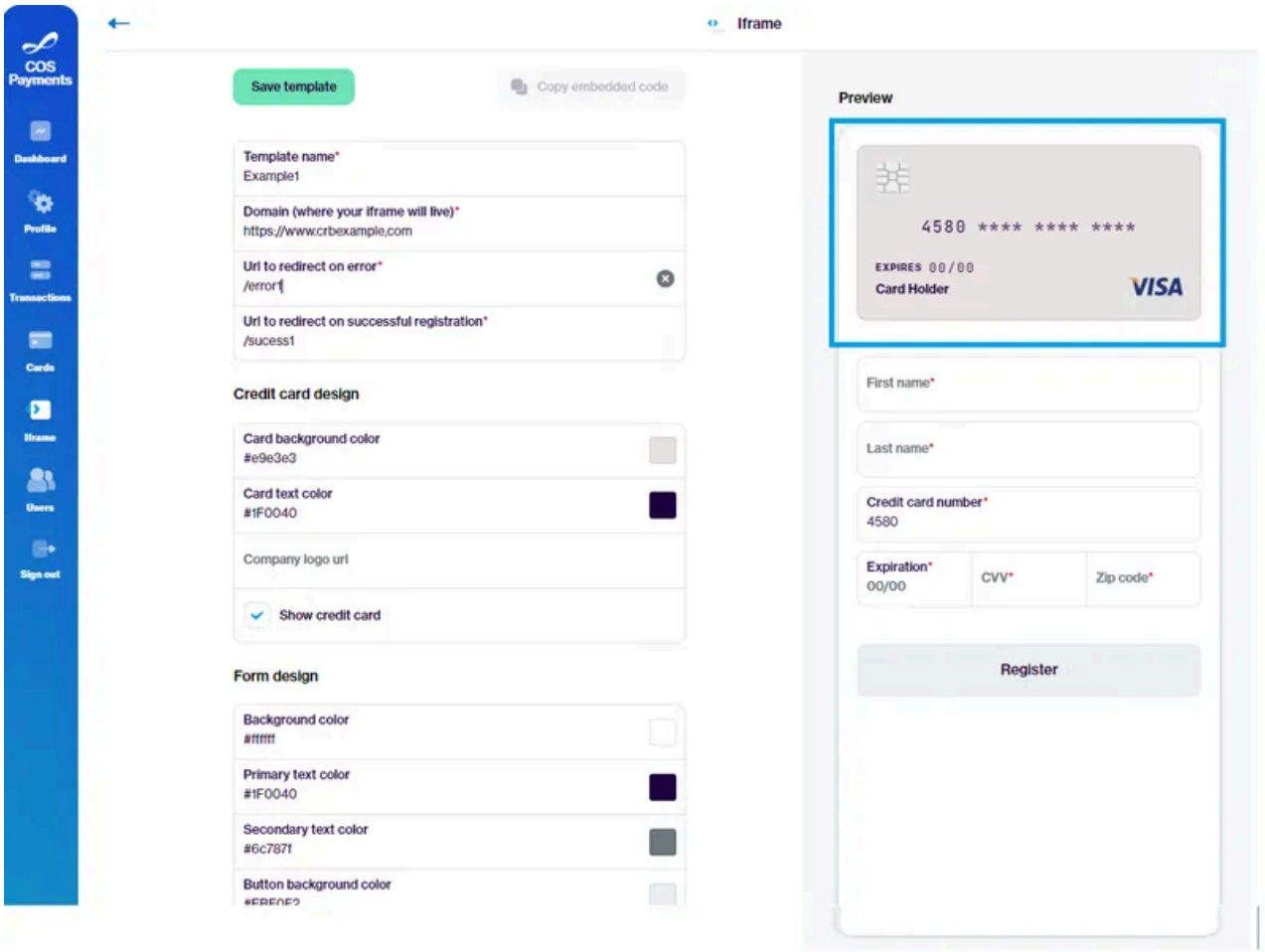
1 Add a template

1. Launch the **Customer Portal UI**.
2. In the left-side icon bar, click **iFrame**.
3. In the iFrames window, click **+ Add template**.



2 Define a style

1. Assign the template a name and domain. Define registration and error paths.
2. You can use the **Credit card design**, and **Form design** sections to customize your iFrame look and feel.
3. Click **Save template** when you're done.



3 Build a URL

1. Click **Copy embedded code**.
   A detailed pop-up tells you how to:
   a. Build the iFrame URL using the **generate an iFrame OTC signup** API.
   b. Insert the API-generated parameters into the iFrame source code.
   c. Paste the iFrame code into your merchant site HTML.
2. Click **OK, got it**.

Now your iFrame is ready to sign up cards.

1. Use the following api together with the idToken to get the params to build the iframe url.

```
Url: https://p2pcustomerportalsandbox.crbcos.com/api/Iframes/IframeOtc
Method: POST
Body: {
  domain: "https://www.theclientgoogle.com",
  templateId: "92f9c06a-84c6-4c62-ab69-08dbe8fdbff0"
}
```

2. Insert the params into the following iframe src:

```
<iframe src="https://p2pcreditcardiframesandbox.crbcos.com/?{response.result.params}" width="100%" height="100%" frameborder="0" scrolling="no"></iframe>
```

3. Paste the iframe code into the html of your website.

Note: The URL will expire in 5 minutes. To enhance the users expirience repeat the steps to genrate a new OTC before the time expires.

OK, Got It

4.3. International payments

International payments APIs

Cross River delivers International Payments (cross-border money movement) which enable you to send money internationally, leveraging SWIFT and local in-country rails. Cross River facilitates this through access to a correspondent banking network.

What Cross River offers

- 24/5 executable payment quotes with superior foreign exchange rates
- Real-time updates on a payment from quote to delivery
- Industry-leading compliance and BSA/AML integrations
- **SWIFT and in-country local rail** options
- Built-in fee customization to earn revenue on each transaction

Currently, Cross River offers **outbound capabilities**. Our solution is payment type agnostic, routing the transactions through the most relevant rails depending on corridor, transaction size, currency and so on.

> For each country you can send any of its supporting currencies. However, the beneficiary account must be set up to receive the currency sent.

Our reach includes these countries:

Australia	Austria	Belgium	Canada	Estonia
Finland	France	Germany	Greece	Ireland
Italy	Japan*	Lithuania	Luxembourg	Netherlands
New Zealand*	Norway	Portugal	Slovenia	Spain
Sweden	United Kingdom			

** Only available using SWIFT and not using local rail*

Available currencies include:

$ Australian dollar (AUD)	$ Canadian dollar (CAD)	€ Euro (EUR)	¥ Japanese yen (JPY)
$ New Zealand dollar (NZD)	Kr Norwegian krone (NOK)	£ Pound sterling (GBP)	Kr Swedish krona (SEK)
$ United States dollar (USD)			

> No crypto or gaming currencies are available.

Find out more

- **Benefits and limitations**
- **COS Explorer**

Tutorials

See our tutorial to learn how to:

- **Send an international payment**.

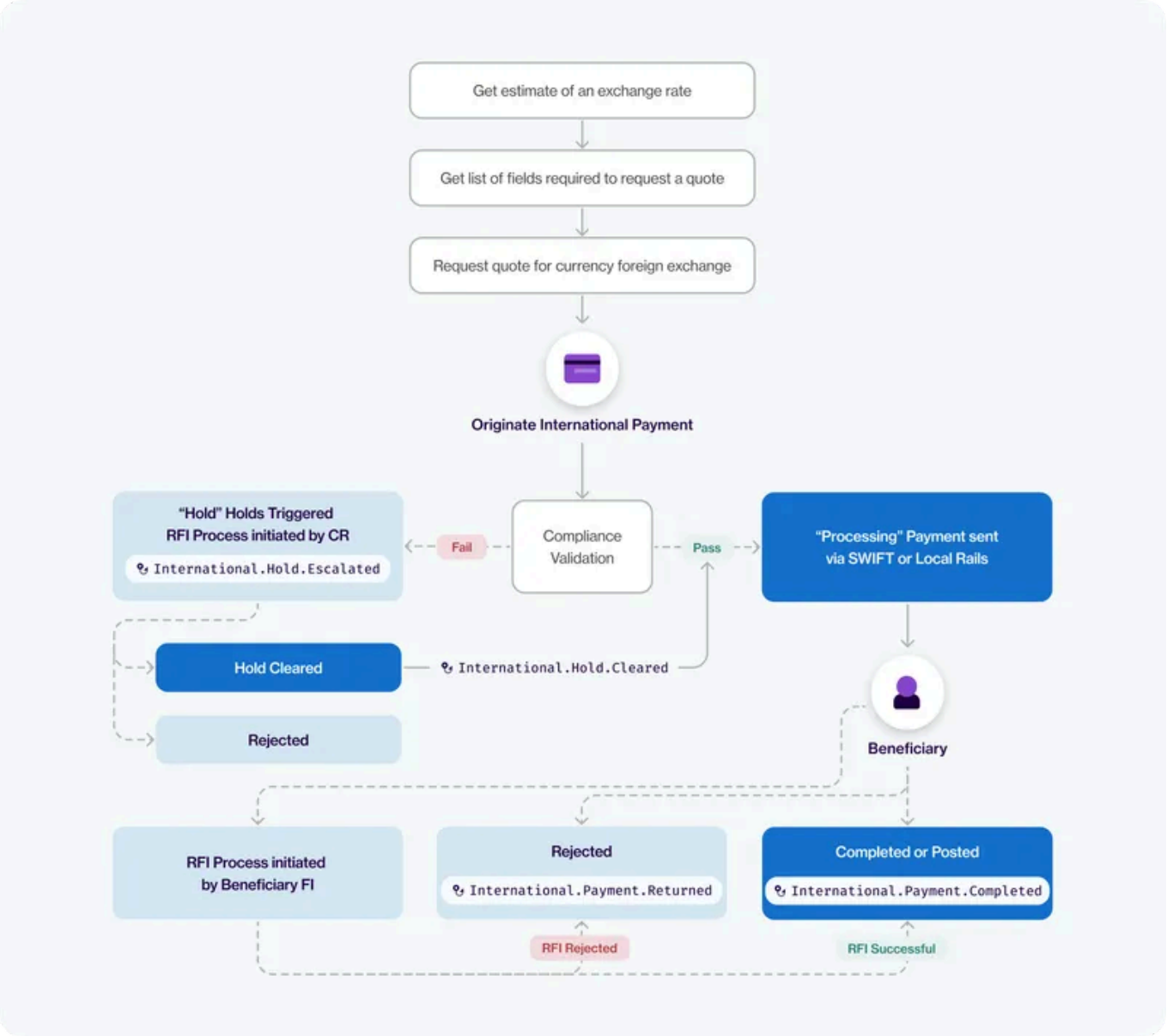
4.3.1. Send payment

Flow of funds

Outbound payments

> See our tutorial to learn how to **send an international payment.**

The following diagram shows the status and webhook flows for outbound International Payments:



How it works

Step 1: Obtain Exchange Rate Estimate

Before committing to sending an international payment, you can **obtain an estimate** of the exchange rate. The estimate includes the current FX rate, with spread and transaction fees. This step does not require beneficiary data.

Step 2: Verify Requirements for Quote

Once satisfied with the exchange rate estimate, you can use the **get list of required fields for quote request** API to validate the specific attributes/information required for creating a valid quote. This step is crucial as international payment requirements vary by country.

Step 3: Generate Executable Quote

After gathering the necessary requirements for a quote, you **request a payment quote**. The quote includes the transfer cost, fees, and the exchange rate for a foreign currency. Beneficiary details are required, and the quote is valid for a limited amount of time (typically 30 seconds). The generated quote ID is essential for the next step.

Step 4: Originate International Payment

Using the obtained quote ID, you use the **originate a payment** endpoint to initiate an international payment. The payment process involves clearing holds, acceptance of conversion, and eventual completion. You are notified via a **webhook** about the acceptance or rejection of their payment.

> Some payments may trigger holds or require additional information from the recipient bank, initiating an RFI (Request for Information) process. Completion of the RFI process is necessary for payment completion. Customers are notified of payment rejection, if any, with reasons provided via a webhook.

4.3.2. SWIFT or local rail

SWIFT

SWIFT, the **Society for Worldwide Interbank Financial Telecommunication**, makes sending international payments easier by providing a secure messaging system between financial institutions.

When you send a payment by SWIFT it settles quickly, typically within T+0 or T+1. However, because intermediary financial institutions along the payment route may charge handling fees or commissions, SWIFT has additional charges. These charges are not always the same. The amount depends on the specific financial institutions involved in the transaction.

Local rail

When you pay through a local rail, such as SEPA (Single Euro Payments Area), typically you'll have no handling fees or commissions, because this payment rail uses banks in the recipient country to make international payments easier. Payments sent by local rails settle efficiently, typically within T+0 to T+2.

> **IMPORTANT**
> With both SWIFT and local rails the payee receives the full amount. The sender pays any fees or commissions.

For certain countries your only option is to use SWIFT because no local rail is accessible. The table of countries at the top of this page shows which countries are limited to SWIFT transactions.

In the **International payments** API (/International/v1/meta/quote-requirements and /International/v1/quotes) , set `priority` to **TRUE** to transfer funds using the SWIFT network. Set `priority` to **FALSE** to use a local rail in the recipient country.

4.3.3. Benefits and limitations

International Payments introduces a new dimension to our bank payments offering, enabling you to send and receive international payments.

Benefits

International payments with Cross River provide:

- **Expanded reach**: International payments seamlessly connects you to a robust network of correspondent banks worldwide, expanding our reach to 22 countries and accommodating 9 currencies.
- **Integration simplicity**: International payments effortlessly integrates with our existing domestic payments framework, creating a plug-and-play solution for outbound payments across the globe.
- **Versatile processing options**: International payments are available through API integration, interface, and file ingestion. You simply provide payment information, and we take care of the entire end-to-end processing.
- **Cost efficiency**: You don't incur a domestic wire fee as a part of your transaction, enhancing cost-effectiveness.
- **Accelerated processing**: International payments ensures faster processing compared to using a domestic partner bank, streamlining the payment workflow for increased efficiency.
- **Fast delivery for key currencies**: Certain currencies, such as Euro, GBP, and CAD, can be delivered as early as the same day and guaranteed for next-day delivery, ensuring timely and efficient transactions.

Limitations

The following limitations currently exist for International payments at Cross River:

- We only provide outbound capabilities. In the future we'll support inbound payments as well.
- Exclusively for entities based in the United States.

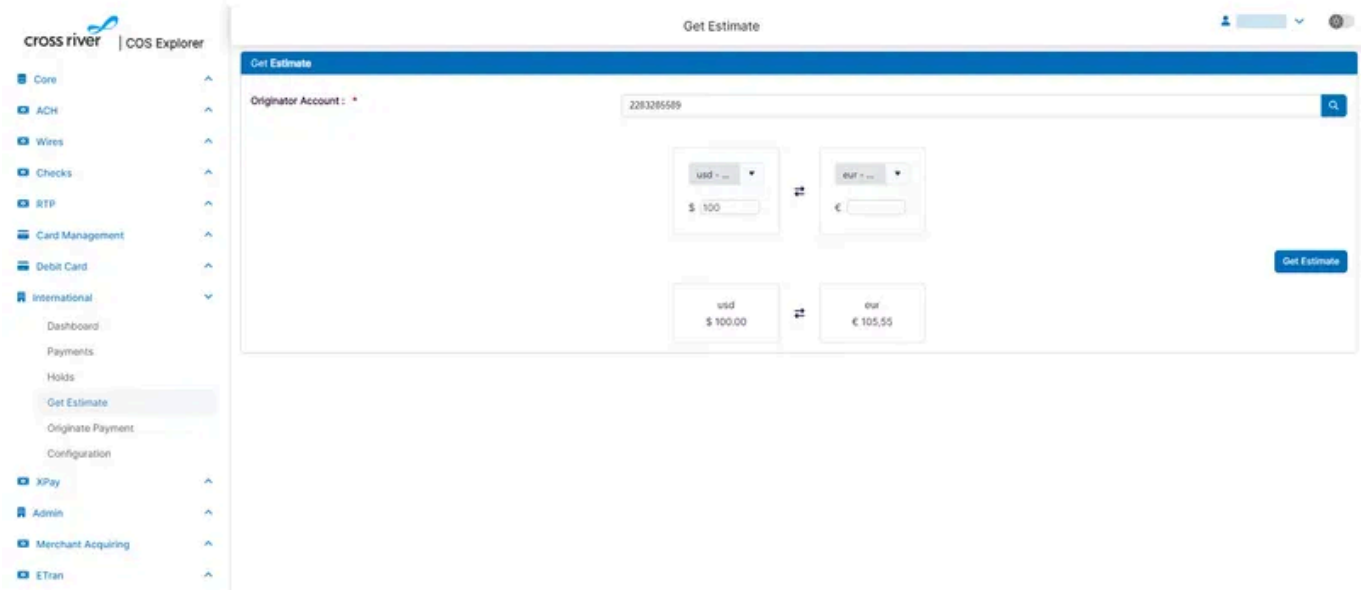
- Only used by business entities (payers), covering both Business-to-Consumer (B2C) and Business-to-Business (B2B) scenarios.
- No support for transactions related to cryptocurrencies or gaming activities.
- No foreign exchange (FX) wallet holding capabilities.
- No remittance payments.

Get estimate

Before committing to sending an International Payment, you can get an estimated FX rate for your desired currency. This optional step includes Cross River's spread fee charge and does not require beneficiary information.

In COS Explorer, go to the **International** tab:

1. Click **Get Estimate**. The **Get Estimate** page displays. Enter your account number (**Acct #**) in the search bar.
2. In the currency box enter *either* the dollar amount you want to send *or* the foreign currency amount you want the beneficiary receive. Do not enter both values.
3. Click **Get Estimate**. A currency estimate displays.



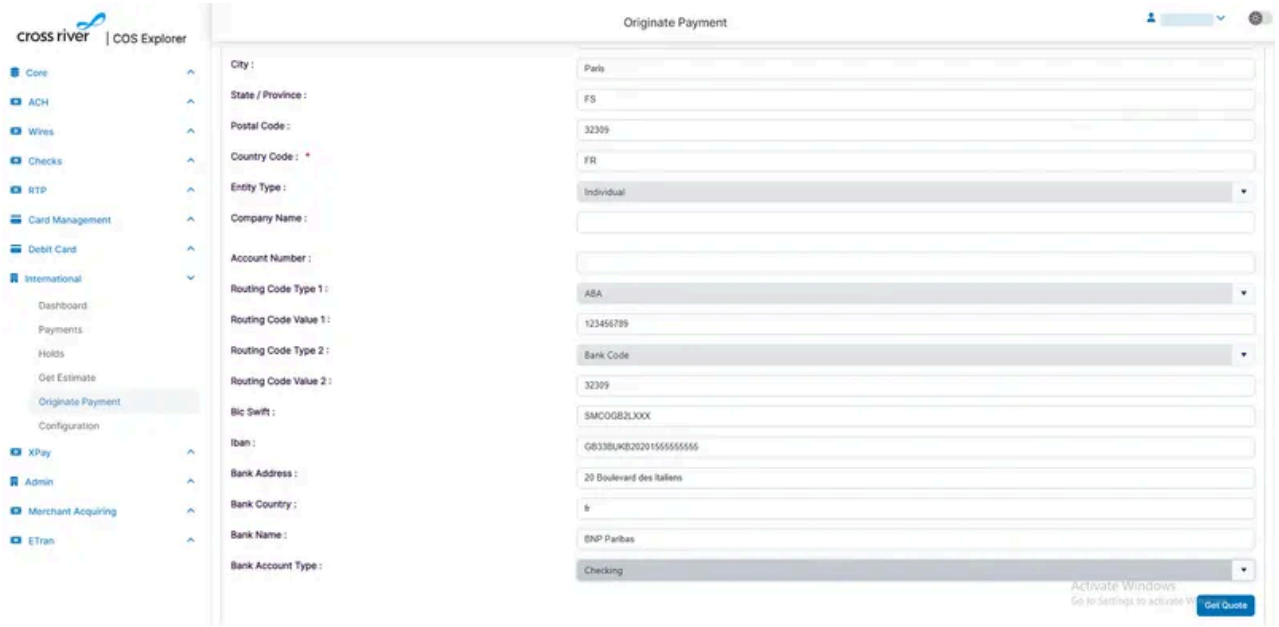
Originate an international payment

1. Click **Originate Payment** and complete all required fields including beneficiary details and **Priority**, where **Yes** indicates **SWIFT** and **No** indicates **local rail**.

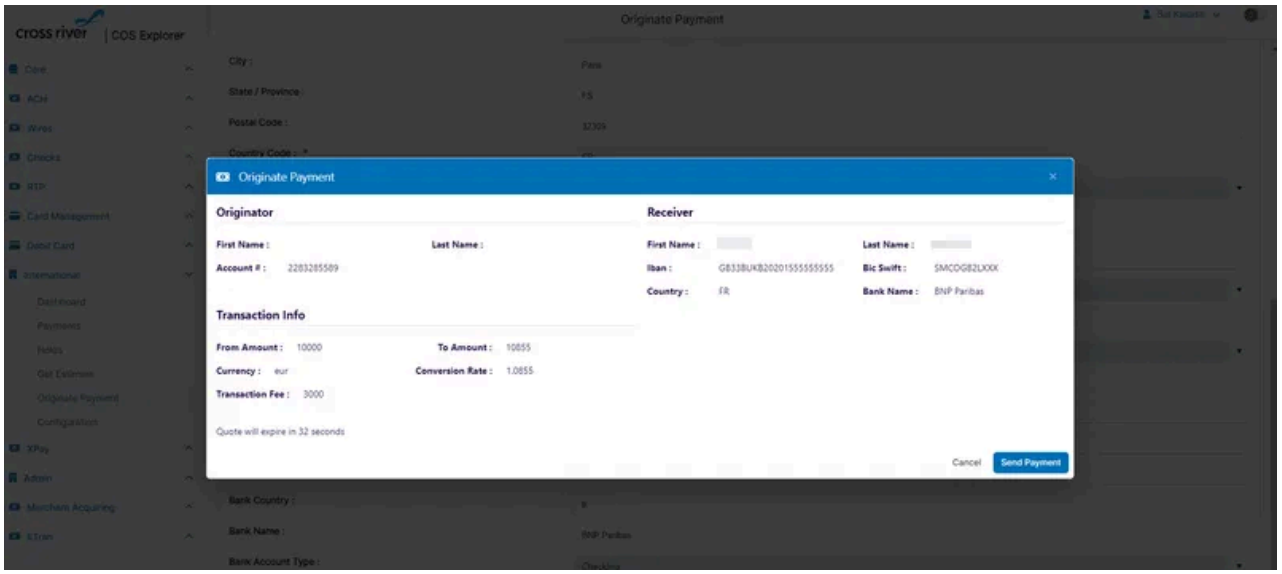
> The required information, such as, account number, IBAN, BIC/SWIFT, sort code, may vary depending on the destination country of the international payment.

2 Click **Get Quote**.

3 After clicking **Get Quote**, you will see the FX rate, spread fees, and transaction fees for the payment. The FX quote is valid for 1 minute before it updates. If you click **Send payment** after the quote expires you will need to go back to Step 2 and regenerate a quote.



4 Click **Send Payment** to originate the payment.



5 To view your payment click **Payments**.

4.4. ACH

ACH APIs

Cross River delivers payments through the Automated Clearing House (ACH), a common electronic payment method usually used for direct deposits of paychecks and bill payments. ACH is governed by **Nacha**, the National Automated Clearinghouse Association.

When you transfer funds by ACH via Cross River, you quickly originate and receive domestic ACH payments either as a single payment or in batches. We provide payment validation through our platform before releasing payments to the Federal Reserve.

What Cross River offers

- Settlement Priority: **Same day** and **Standard** settlement options. *Same Day* transactions are processed and settled on the same business day while *Standard* are processed and settled the next business day. Credit transactions may be processed up to two business days prior to settlement day.
- **Batched payments**: You can send multiple payments in a single API call.

Find out more

- **Send and receive**
- **Payment types**
- **Participant roles**
- **Payment settlement**
- **Time windows**
- **SEC codes**
- **ACH in COS Explorer**

Tutorials

See our tutorials to learn how to:

- **Send an ACH payment**
- **Send a client batch**
- **Simulate inbound ACH payments**

4.4.1. Send and receive

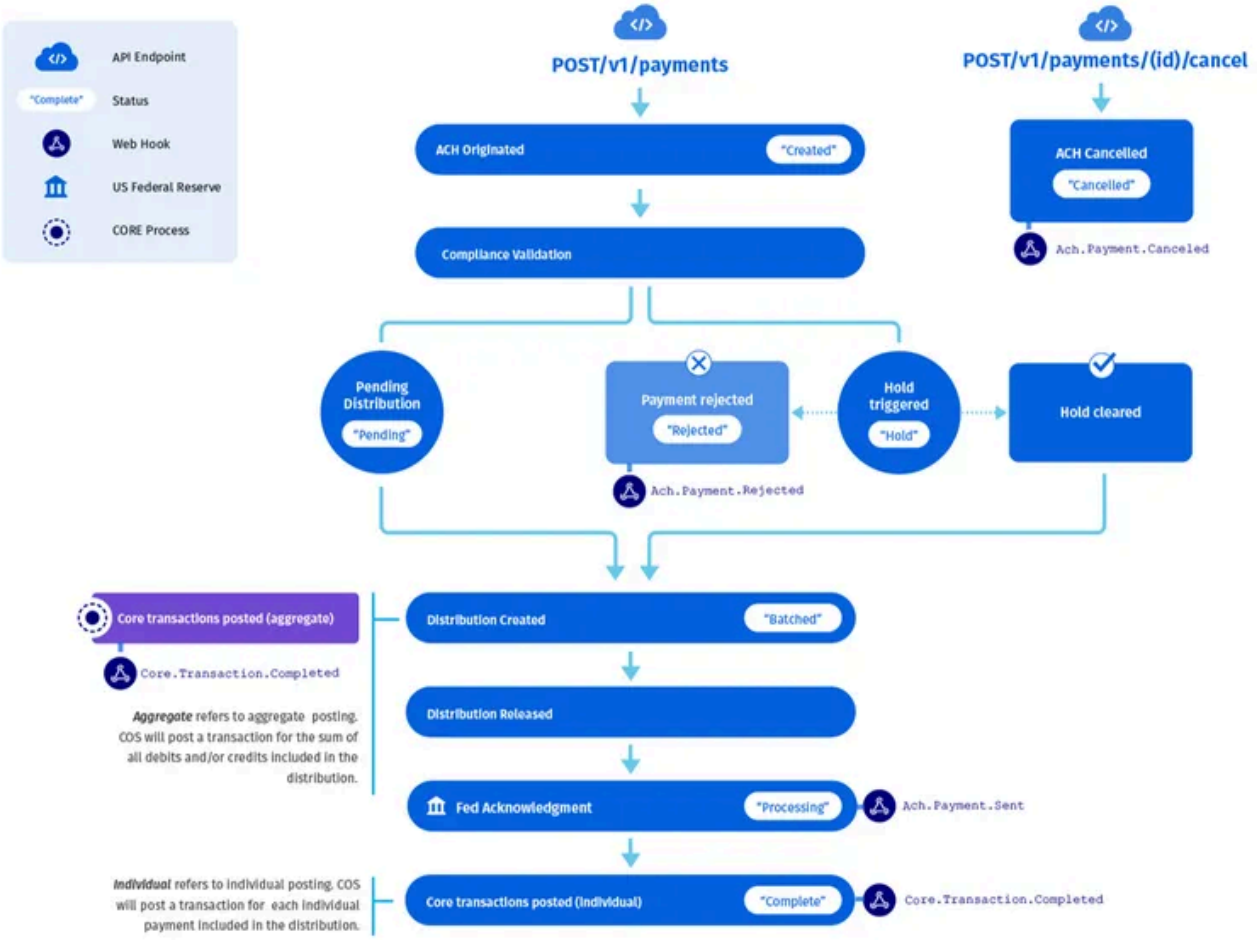
ACH allows you to transfer funds electronically and cost-efficiently through the Federal Reserve ACH network. Cross River offers both outbound and inbound ACH payments, enabling you to both originate and receive ACH payments. Cross River supports both **Same Day** and **Next Day (standard)** ACH settlement.

> See our tutorial to learn how to **send an ACH payment**.

Outbound

When a customer originates an Outbound ACH payment, funds are either pushed to or pulled from an account at an external bank. An *Outbound Push* sends funds to the recipient's account and this is a credit to the receiver. An *Outbound Pull* submits a request to pull funds from an external account to the account at Cross River and this is a debit for the originator.

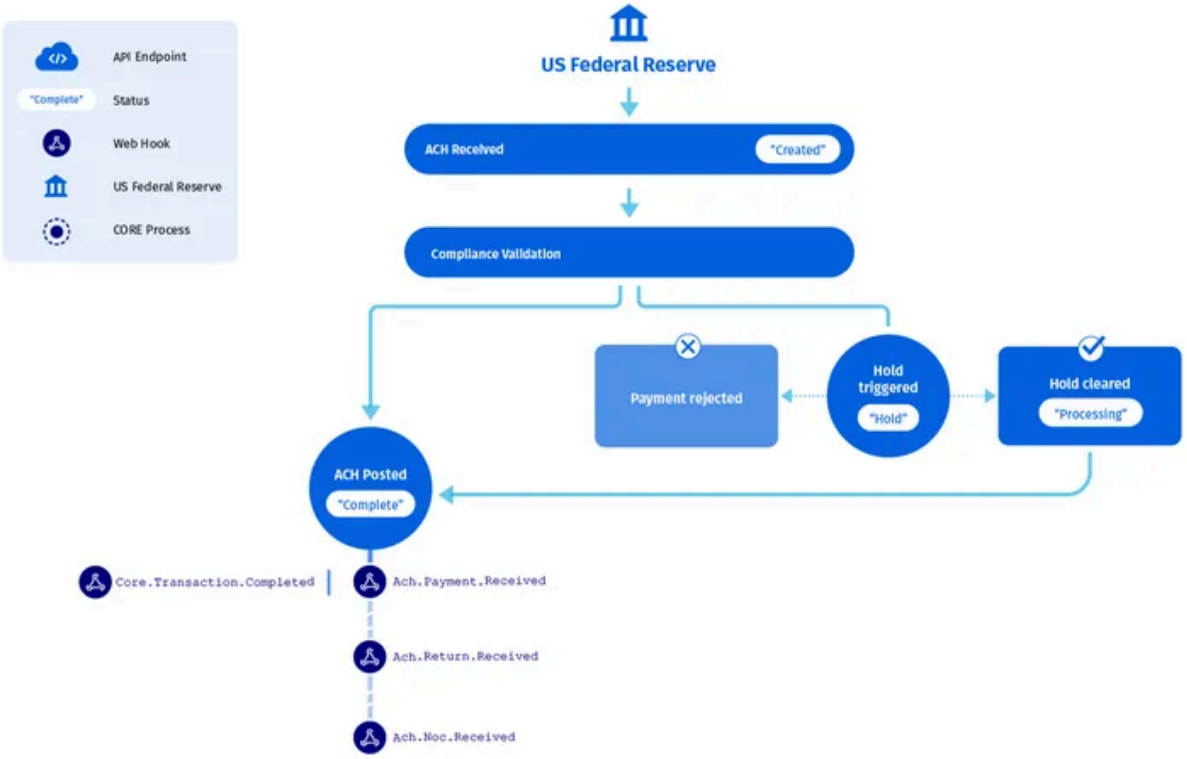
The following diagram shows the **outbound flows** and **webhook events** for ACH transactions.



Inbound

An Inbound ACH payment involves receipt (push) to or authorized withdrawal (pull) from an account at Cross River. An *Inbound push* involves a customer at an external bank sending payment to an account at Cross River and this is a credit for the receiver. An *Inbound pull* involves a customer at an external bank pulling money from an account at Cross River and is a debit to the originator.

The following diagram shows the **inbound flows** and **webhook events** for an ACH payment.


API Endpoint
"Complete"
Status
Web Hook
US Federal Reserve
CORE Process
US Federal Reserve
ACH Received
"Created"
Compliance Validation
Payment rejected
Hold triggered
"Hold"
Hold cleared
"Processing"
ACH Posted
"Complete"
Core.Transaction.Completed
Ach.Payment.Received
Ach.Return.Received
Ach.Noc.Received

4.4.2. Payment types

Cross River offers both Standard and Same Day ACH payments. You can send payments using our API either individually, or in **batched payments**.

> **IMPORTANT**
> The payment information and data you send to Cross River must include a **Nacha SEC code** and conform to **Nacha settlement rules**.

Standard ACH

Use standard ACH payments for regular funds transfers such as paychecks or bill payment. A standard ACH payment can take up to two business days to **settle**. The Fed refers to this as "future dated."

Cross River supports all domestic ACH payment transactions. You can transfer funds from one US bank account to another.The `serviceType` attribute value in the **originate an ACH payment** API will always be **standard**.

An ACH payment is either a push or pull depending on your **role** in the transaction. Both the originator and the receiver can push or pull funds.

- The Originating Depository Financial Institution (ODFI) sends fund or sends a pull requests (to receive funds)
- The Receiving Depository Financial Institution (RDFI) receives funds or receives a pull request (to send funds)

Set the value to **pull** or **push** for the **transactionType** attribute in the **originate an ACH payment** API.

See our **ACH APIs** and our **ACH originate a payment** tutorial.

International payments

> For the Cross River comprehensive international payment solution, see **International payments**.

The ACH system is domestic (US only). However, it can facilitate funds transfers through similar payment systems in other countries. This allows you to transfer funds internationally using ACH.

If you want to transfer funds internationally using Cross River and ACH, you need a US intermediary FI (a US bank that can do international transfers) with a relationship with the FI of the international account. We pass the payment information and funds to the intermediary FI, which sends it to the corresponding international account.

If you want to receive funds from outside the US using ACH, the funds will first be transferred to an intermediary FI, and from there be transferred to us.

Any payment over the ACH network that originates or ends in a foreign FI must use the **IAT SEC transaction code**. The Cross River service type will always be **Standard**.

Validation (prenote)

You can use an *ACH prenote* if you want to validate bank account information before sending or receiving any funds.

For example, if a utility company automatically withdraws a payment from a customer account, it sends an ACH prenote before it sends the request to withdraw the funds. A successful prenote call results in no error or change notification.

Sending an ACH prenote is simple. You **originate the payment** with a value of **$0.00**. You can use ACH prenotes for both credits and debits.

Learn more about ACH prenotes and account validation from the Nacha **Validation Resource Center**.

Same day ACH

When you send a payment using Same day ACH, the funds are available to the RDFI on the **settlement date**. The payment must be processed by FedACH before 4:45pm EST for it to be settled on the same day.

There is a limit of $1,000,000.00 per Same day ACH transaction.

Same-day ACH uses the **same endpoint** as Standard ACH, however the **time window** is smaller and you set the `serviceType` attribute to **SameDay**.

> **IMPORTANT**
> The payment information and data you send to Cross River must conform with the **Nacha SEC codes** and **Nacha settlement rules**.

4.4.3. Participant roles

An **ACH payment transaction** can be complex, involving a number of individuals and financial institutions. The table below describes participants involved in a typical transfer of funds.

Participant	Explanation
Originator	The party - individual or company - that originates a **credit, debit, or non-monetary entry** into the ACH payment system for receipt by the Receiver account. The Originator must have an arrangement with the ODFI directly or through a third-party sender and an authorization from the Receiver to initiate the transaction.
Originating Depository Financial Institution (ODFI)	The financial institution that originates the payment and forwards it to the ACH Operator.
ACH Operator	Receives entries from ODFIs, sends the entries to appropriate RDFIs, and performs settlement functions for the financial institutions. Cross River works with the **Federal Reserve Bank (FedACH)**.
Receiving Depository Financial Institution (RDFI)	The financial institution that receives the credit or debit entry from the ODFI and posts it in the receiver account.
Receiver	The party (individual or company) that receives the debit or credit entry.
Third-Party Service Provider (optional)	Acts on behalf of the originator, ODFI, or RDFI regarding ACH entries and transactions. This includes creating ACH files for an originator or ODFI and acting as a sending or receiving point for an ODFI or RDFI. A third-party sender acts on behalf of the originator when there is no agreement between the ODFI and originator for ACH origination services. A third-party sender is never the originator for entries it transmits on behalf of another organization. A third-party sender of entries may be an originator of other entries. Cross River can act as the ODFI for outbound ACH transactions or the RDFI for inbound ACH transactions.

> A Depository Financial Institution (DFI) can act as an RDFI without being an ODFI. If a DFI wants to originate ACH entries, it must act as both the ODFI and the RDFI. The ODFI must be able to receive payment returns.

All parties in an ACH transaction must comply with the **Nacha rules**.

4.4.4. Payment settlement

- **Standard ACH** transactions, whether debit or credit, will typically settle within 1-3 days for processing and clearing. Some transactions, especially those initiated through
- **Same Day ACH** settle on the same day. Factors determining settlement time include: if the transaction is Same day vs. Next day, transfer type such as **credit** vs. **debit**, **time of day** due to batch processing, and the Federal Reserve operating hours.

Credit entries

An ACH Credit occurs when a business or individual (originator) authorizes their bank (ODFI) to send money to a receiver account at another financial institution. The RDFI posts entries and provides funds availability.

Posting entries and funds availability are determined by the *Settlement Date* in the Company Batch Header Record.

Standard ACH credit entries

Standard ACH credits are made available to the RDFI by the ACH Operator no earlier than two banking days prior to the Settlement Date. It is recommended that credits be posted or at a minimum, memo-posted, prior to opening of business on Settlement Date. However, credit entries can be posted prior to the Settlement Date if the RDFI cannot warehouse the entry. The RDFI should consider the effect of posting credits and making funds available to the Receiver prior to the date on which the RDFI receives settlement for those credits.

The Nacha Operating Rules require that credit entries be made available for withdrawal by the customer on the Settlement Date. An RDFI can satisfy this obligation by making the entry available for withdrawal at the RDFI teller facilities by the close of business on the Settlement Date or at an ATM by midnight on the Settlement Date.

The RDFI must make funds available for withdrawal or cash withdrawal for any standard processing credit entry:

- No later than 9 am. (RDFI local time) on the Settlement Date.
- The entry is made available to the RDFI by the ACH Operator no later than 5:00 pm. RDFI local time on the banking day prior to the Settlement Date.

This rule applies to all credit entries that meet the timing requirement, regardless of Standard Entry Class Code. An entry or entry data is made available to an RDFI or its receiving point when the entry or entry data is processed by the RDFI ACH Operator and is ready for distribution.

Same-day processing ACH credit entries

Same-day processing ACH credits will be delivered to the RDFI on the Settlement Date.

- RDFIs must generally make available for withdrawal the amount of a same-day credit entry received in the first same-day processing window.
- They must do this no later than 1:30 pm. RDFI local time on the Settlement Date.

An RDFI that reasonably suspects that a credit entry is unauthorized is exempt from these requirements, subject to applicable legal requirements. An RDFI invoking such an exemption must promptly notify the ODFI.

For a same-day credit entry received in the second same-day processing window, most RDFIs are required to make funds available for withdrawal no later than 5:00 pm. RDFI local time on Settlement Date.

Debit entries

Debit entries are the opposite of credit entries.

Standard processing ACH debit entries

Standard processing of ACH debits will be made available to an RDFI no earlier than one banking day prior to the Settlement Date.

Same-day processing ACH debit entries

Same-day processing of ACH debits will be made available to the RDFI on the Settlement Date. The Nacha Operating Rules prohibit RDFIs from posting debits prior to the Settlement Date.

ACH payments only settle on business days (no settlement on weekends or federal holidays). Payments only occur on business days (no support on weekends or banking holidays).

Nacha settlement rules

Nacha has its own set of limitations and restrictions:

- According to Nacha rules, settlement happens the day after the payment is initiated.
- Nacha operating rules require standard ACH *credits* to settle in 1-2 business days.
- ACH *debits* settle the next business day.
- There is no settlement on weekends or federal holidays.
- *Same day* settlement happens on the same day as the payment is originated.
    - The entry must be processed by FedACH before 4:45pm EST for it to be settled on the same day.
- Domestic payments must be less than $1,000,000
- Same day ACH payments are not eligible for international transactions
- Payments can take up to 90 days to finalize.
  An ACH payments may be rejected because of insufficient funds or because it is unauthorized, for up to 90 days after the date of the transaction.

4.4.5. Time windows

When **sending an ACH payment**, it's important to understand how the time of day the transaction is sent and the `serviceType` value impact when the funds actually arrive in the account. You need to be aware of the **Cross River time windows**.

Payment effective date

When an ACH payment **settles** in an account:

- Any transaction tied to an ACH payment executes on the effective date of the payment.
- The `serviceType` (either *Standard* or *Same Day*) and the configured cutoff time influence the effective payment date.
- When the payment is released to the Federal Reserve and receipt of payment is acknowledged, we execute the transactions on the payment `effectiveDate`.

See **payment types** and the **send an ACH payment** API.

For example:

- A Same Day payment originates on Tuesday, 8/24 at 12 pm. The cut off time is configured for 1 pm. The `effectiveDate` of the payment is 08/24.
- A Same Day payment originates on Tuesday, 8/24 at 3 pm. The cut off time is configured for 1 pm Same Day cutoff. The `effectiveDate` of the payment is 08/25 because the payment originated after the cutoff time. The service type remains `SameDay` in the system, despite the next-day effective date.
- A Standard payment originates on Tuesday, 08/24. The `effectiveDate` of the payment is 08/25.

Cross River time windows

Cross River batch-sends files received throughout the day. To process a payment for any time window, the files must go through the due diligence process before the cutoff time.

Same day Domestic US	**10:00 AM - 2:00 PM EST**
Standard/Next Day Domestic US	10:00 AM - 01:45 AM EST
Standard/Next Day Domestic IAT*	10:00 AM - 2:00 PM EST

*Dependent on additional processing checks.

Cross River accepts Same day and Standard transactions during the above times in accordance with the FedACH processing schedule. The windows ensure sufficient flexibility in processing and submitting transactions in a timely manner to meet settlement times. Window cutoff times are customizable to meet your processing needs.

4.4.6. Batched payments

The Cross River system supports the ability to send up to 5000 payments in a single request via the **client batch API**. A batch request is always the preferred method of originating multiple payments at a given time.

Once a batch is accepted, we break it into individual payment records that may go on hold or be sent to the Federal Reserve independently. **Webhook events** fire as usual on each of these individual payments. The overall **payment workflow** is identical to that of payments originated individually.

It is important to note that client batch origination requests are asynchronous and the associated payment records may not be available immediately. You can monitor the progress using the API via the `status` and `importCount` fields of the batch object.

> See our tutorial to learn how to **send ACH client batch payments**.

4.4.7. SEC codes

A SEC code indicates the Standard Entry Class required by Nacha for every ACH transaction. When you **originate an ACH payment** by API, you enter this code in the `secCode` field of the request. The table below explains common SEC codes. Learn more about SEC codes on the **Nacha website**.

> Your Operations Support Team will guide you about which SEC code to use based on the use case you support.

Code	Name	Description	Type	Transaction Type	Credit/Debit
ARC	Accounts Receivable Entry	A single debit entry to an account initiated for purchases or payments that are made in person, via US mail, or placing the payment in a dropbox.	Consumer or Non-Consumer Mail Order or Retail	Single Entry	Debit only
BOC	Back Office Conversion Entry	A single debit entry to an account for in-person purchases or payments made at the point-of-purchase. *Check is not returned to check writer.*	Consumer or Non-Consumer Retail	Single Entry	Debit only
CCD	Corporate Credit/Debit Entry	Funds are transferred between unrelated corporate entities or transferred as intra-company cash concentration and disbursement transactions.	Non-Consumer	Single or Recurring Entry	Credit/Debit

Code	Name	Description	Type	Transaction Type	Credit/Debit
		Proof of Authorization for transactions ran on the web using CCD will adhere to the same requirements as a web transaction.	Retail, Phone Order, Ecommerce, and Mail Order	Single or Recurring Entry	Credit/Debit
CIE	Customer Initiated Entry	A single credit entry initiated by a consumer (originator) to a non-consumer account. CIE is a consumer-initiated credit entry, a credit push to a merchant, typically used by a financial institution's or third-party's bill pay service and would be not be Originated by a merchant.	Consumer	Single Entry	Credit only
COR	Notification of Change, or Refused Notification of Change	This Standard Entry Class Code is used by an RDFI or ODFI when originating a Notification of	Consumer or Non-Consumer	N/A	N/A

Code	Name	Description	Type	Transaction Type	Credit/Debit
		Change or Refused Notification of Change in automated format. It is also used by the ACH operator that converts paper Notifications of Change to automated format.			
CTX	Corporate Trade Exchange	Used to transfer funds between a buyer's and a seller's bank accounts.	Non-Consumer	Single or Recurring Entry	Credit/Debit
DNE (Federal Govt. Agency Use Only)	Death Notification Entry	A non-monetary entry from a Federal Government agency. Notifies the Financial Institution that the recipient of a government benefit has passed away.	Consumer	N/A	N/A
ENR	Automated Enrollment	A Non-Monetary Entry that triggers the origination of	Non-Consumer (Federal	N/A	N/A

Code	Name	Description	Type	Transaction Type	Credit/Debit
	Entry	ACH credit or debit transactions to the account holder at the DFI. The ENR process allows DFIs to transmit information to Federal Government agencies, on behalf of their account holders, that serves as enrollment for either ACH credit or debit activity.	Government Agency)		
IAT	International ACH Transaction	Credit or Debit Entry is part of a payment transaction that involves a financial agency's office not located within the territorial jurisdiction of the United States. NOTE: This SEC code is implemented only for very specific	Consumer or Non-Consumer	Single or Recurring Entry	Credit/Debit

Code	Name	Description	Type	Transaction Type	Credit/Debit
		merchants and is not generally available.			
POS	Point of Sale payments	See **POS card transaction types**	Consumer	Single Entry	Credit/Debit
PPD	Prearranged Payment and Deposit Entry	A single or recurring credit transaction for payment of payroll, expense reimbursement, dividends, retirement, interest, etc.	Consumer	Single or Recurring Entry	Credit/Debit
RCK	Re-presented Check Entry	A re-presented check entry with security measures in place.	Consumer	Single Entry	Debit
TEL	Telephone-Initiated Entry	A single or recurring debit transaction initiated orally via the telephone.	Consumer	Single or Recurring Entry	Debit
WEB	Internet-Initiated Entry	**Credit** - A single or recurring credit transaction from the account of a natural person to the account of a natural person. Cannot be used	Consumer	Single or Recurring Entry	Credit or Debit

Code	Name	Description	Type	Transaction Type	Credit/Debit
		for business-to-consumer transactions. **Debit** - A single or recurring debit transaction initiated during a secure (minimum 128-bit encryption) internet or mobile session.	Ecommerce		

4.4.8. Early direct deposit

Early directGenerally, when an employer deposits a paycheck directly into an employee bank account, the money only becomes available to the employee up to 2 days later.

Cross River Early Direct Deposit (EDD) gives your customer earlier access to their money if they receive their paycheck directly to their bank account. Cross River gives your customers EDD functionality for payments sent via **ACH**. Funds settle when Cross River gets the ACH file from the Federal Reserve, instead of waiting to release the money to the account on the payment effective date. When you apply EDD to a specific product type, all your customers using that product are eligible.

> While EDD is primarily used for paycheck deposits, any deposit amount that falls between 0 and the EDD threshold set for the deposit account product type and has the values shown in the **sandbox testing section** will be treated as an EDD.

Requirements for your customers

Ensure that your customers:

- Understand the requirements and limitations
- Understand that EDD is not guaranteed and depends on timely transfer of ACH files
- Have checking or savings accounts set up with Cross River
- Are enrolled for direct deposit with their employers. Once EDD is enabled, 1 to 2 pay cycles are usually necessary before it takes effect.

Typical flow

1. Your customer's employer instructs their bank or payroll agency to initiate payroll payment.
2. The employer’s bank sends an ACH file to the Federal Reserve.
3. The Federal Reserve sends the ACH file to Cross River.

4. Cross River settles funds to customer's account on the same day the ACH file is received (up to 2 days before the effective date). A memo post* is added to the available balance. On the effective date, the memo post is replaced seamlessly by a full transfer.
5. The customer can access funds upon settlement.

***Memo posts** indicate that funds will go out or come in and affect the available balance accordingly. For example, with debit cards the available balance drops. In this case, the available balance increases.

Enable early direct deposit

Reach out to your relationship manager to enable Early Direct Deposit. Once you sign a contract addendum, COS Ops enables EDD for your customers.

Limitations

Be aware of the following limitations:

- EDD is not guaranteed and depends on timely transfer of ACH files.
- The customer's employer must initiate the payroll payment on time, and the employer's bank or payroll agency must ensure delivery of ACH files to the Federal Reserve before the effective date.
- The amount sent must be greater than 0 but less than the early ACH threshold configured for your product.

Test in the sandbox

Once we enable EDD for you in the sandbox, call the simulate ACH inbound payment endpoint to generate an ACH credit transfer that matches the EDD requirements.

1. **Register** for the `Ach.Payment.ReceivedEarly` webhook.
2. Use the ACH `POST /v1/payments/simulated-inbound-originations` endpoint with these values for the following attributes:
    - `transactionType` = **Push**

- `secCode` = **PPD**
- `serviceType` = **Standard**

3. Make sure the amount you send is greater than 0 but less than the EDD payment threshold configured by Cross River for your deposit account product.

4.4.9. ACH in COS Explorer

ACH is a payment method used in the United States for either a debit (pull) or credit (push) transactions between a sender (originator) and recipient (receiver). The ACH network offers Same Day ACH, so that transactions can clear on the same business day they are initiated.

ACH is governed and administered by the National Automated Clearing House Association (Nacha).

To find out more information about ACH at Cross River see the **ACH overview**.

Manage your ACH activities from the **Payments** screen in the **ACH** tab.

ACH payment details

Click a row in the **Payments Search** list.

Payment information displays in the **Payment Details** area.

ACH holds

The ACH Holds tab displays information on any holds on the ACH payment:

Column	Description
STATUS	Status of hold placed on payment
TYPE	Type of hold placed on payment
DESCRIPTION	Description of hold
CLEARED	
REASON	Reason for hold
NOTES	Notes on hold

Place hold

1. Click IMAGE and then select **Place Hold**
2. In **Place Manual Hold** dialog enter information in the **Description** box.
3. Click **Place Hold**., and then select **Place Hold**.

View receipt

1. Click IMAGE, and then select **View Receipt**.
2. **Validation Receipt** displays in the dialog.
3. Click **Dismiss** to close.

Memo posts

The Memo Posts tab displays information on memo posts for an ACH payment:

Column	Description
DATE	Date of memo post
TRACE #	Unique core trace number of transaction associated with the memo post
SUB LEDGER	Sub ledger account number memo post applies to
DESCRIPTION	Description of memo post
CODE	Transaction code of payment
AMOUNT	Payment amount

Related transactions

Click IMAGE and enter the relevant search parameters.

If your ACH configuration is set to aggregate, individual transaction details will not display.

Click in the dialog to see the **Transaction Details**

Column	Description
TRACE #	Unique core trace number of transaction associated with the memo post
STATUS	Status of transaction
DATE	Date of memo post
DEBIT ACCOUNT	Debit account number
CREDIT ACCOUNT	Credit account number
CODE	Transaction code of payment
DESCRIPTION	Description of memo post
RAIL	Rail used for transaction
AMOUNT	Transaction amount

Payment actions

Cancel payment

For a pending payment:

1. Click **Cancel Payment**. The **Cancel Payment?** dialog displays:
2. **Are you sure you want to cancel this payment? This action cannot be undone. Note:** Clicking **Cancel** in the dialog will cancel the payment.
3. Click **Cancel**/**Confirm**.
4. If applicable, enter information in either the **Internal Only Note** or **Partner Note** fields, or both. Select **Same as Internal Note** to use the same text for both notes.

Nacha view

The search results display in the **Transaction Nacha View** dialog.

Column	Description
Batch Header	
SEC	Standard Entry Class(SEC) for payment
Company Name	Company name of originator
DESC	Description
Effective Date	Effective date of payment (YYMMDD)
Orig Status Code	Originator status code

Column	Description
Batch #	Batch number
Desc Date	Date the description was entered
Settlement Date	Settlement date of payment
DFI Identification	Deposit Financial Institution(DFI) identification
Entry	
Transaction Code	2-digit code identifying the account and transaction type
DFI Account	Receiver account number
Identification	Receiver identification , corresponds to Nacha **Individual Identification Number** field
Discretionary	Corresponds to **Discretionary Date** field
Trace #	Unique core trace number of payment
Receiving DFI	Receiver routing number
Amount	Payment Amount
Receiver Name	Name on receiver account
Addenda Indicator	Whether addenda information exists
Nacha	Nacha message for payment

View webhooks

Click IMAGE to toggle the search fields and enter your search criteria. Click **Search**.

Column	Description
EVENT	Payment event
STATUS	Expired, Failed, Pending, Success
CREATED	Time and date of event creation
ATTEMPTED	Time and date of event attempt

Click IMAGE to open the **Webhook Event Dialog**.

Click IMAGE to open the **Webhook Event Details** dialog.

Click **resource** to display further details.

Column	Description
STATUS	Expired, Failed, Pending, Success
ATTEMPTED	Time and date of event attempt
CALLBACKURL	The API call URL
ELAPSED	Time elapsed in milliseconds
CODE	Accepted/ Rejected
IS RESEND	Yes / No

Click **Callback url** to view **Log Details**.

View notes

This dialog displays any **Notes** on ACH payments:

Field	Description
DATE	Note creation date
SUBJECT	Note subject
AUDIENCE	Note audience (Internal / Partner)
CREATED BY	COS user who created the note
Body	Note text

You can also add a note as follows:

1. Click **Add Note**
2. Enter the information into the fields in the **Create New Note** dialog.
3. Click **Submit**.The new note displays in the **Notes** dialog.

Payment history

Click the bottom item in the **Payment History** area to view the **Outbound** or **Inbound Origination** history information for that related payment.

In the **Client Batch Search** screen, click an entry to view a particular batch.

Column	Description
Ref ID	Transaction reference ID
TRACE #	Nacha trace number
STATUS	Payment status
POSTING	Inbound /Outbound
TYPE	Code for posting transaction
PARTNER	Date and timestamp of payment
ORIGINA	Payment originator
RECEIVER	Payment receiver
SEC	SEC Code
TRANS	Transaction (Pull / Push)
AMOUNT	Dollar amount

Client batch tasks

Cancel batch

1. Click **Cancel Batch** to open the dialog.
2. Click **No**/**Yes**.

Client batches

In the **Client Batch Search** screen, click an entry to view a particular batch.

Details of the client batch displays:

Learn more about **client batches**.

Column	Description
Ref ID	Search for client batch by ID
Account	Account number
Partner	Partner name
Payment Count	The number of unique payments in the batch (up to 5000)
Debit Total	The total amount to be credited to the originator account
Created	Time and date batch created (EST)
Imported	Time and date batch imported (EST)
Status	Processing, Imported, Canceled
Product	Partner product
Import Count	How many payments have been processed at the time of the query
Credit Total	The total amount to be debited from the originator account
Last Modified	Time and date batch last modified

Column	Description
Ref ID	Transaction reference ID
TRACE #	Nacha trace number
STATUS	Payment status
POSTING	Inbound /Outbound
TYPE	Code for posting transaction
PARTNER	Date and timestamp of payment
ORIGINA	Payment originator
RECEIVER	Payment receiver
SEC	SEC Code
TRANS	Transaction (Pull / Push)
AMOUNT	Dollar amount

Client batch tasks

Cancel batch

1. Click **Cancel Batch** to open the dialog.

2. Click **No**/**Yes**.

Holds

Column	Description
Ref ID	Payment Ref ID
Partner	Partner the payment is associated with. **Note:** You can click the partner name to view partner details.
Status	Payment status. Pen icon displays when a NOC (Notification of Change) was sent for the payment. Arrow icon displays when the payment has a linked ACH return.
Trans Type	Transaction type (Push/ Pull)
Completed	Time and date of hold completion
Reason Code	Reason code linked to ACH return or NOC
SEC	SEC (Standard Entry Class) code of payment
Direction	Inbound/ Outbound
Service	Service type (SameDay/Standard)
Created	Dateand time the payment was initiated
Source	Paymentsource. For Inbound payments the value will always be “File”, and for outbound payments the value will always be “API” or “Partner Portal”.

Column	Description
Hold Days	Numberof days the credit to an account for an outbound pull payment is held. **Note:** This field applies to outbound pulls only.
Client Identifier	Use this attribute to add your own unique identifying string to a payment call or COS record. This attribute is useful for **idempotency purposes**.
Product	Product associated with payment
Posting	Posting status of payment and account number associated with payment. **Note:** You can click the account number to view the account details. Box icon indicates a single posting
Trace #	Unique core trace number of payment
Data	Company discretionary data on payment
Type	Payment type
Direction	Payment direction (Inbound / Outbound)
Amount	Payment amount
DESC	Payment description
Effective	Effective date of payment (YYMMDD)
Purpose	Purpose of payment
Addenda	Addenda information included in payment
Originator	
Routing #	Originator routing number
Account #	Originator account number
Account Type	Originator account type
Name	Originator name

Column	Description
Identification	10-digit identification number for originator
Data	Company discretionary data
Address	Address 1-3 of originator
City	City of the originator
State	State of the originator
Country	Country of the originator
Receiver	
Routing #	Receiver routing number
Account #	Receiver account number
Account Type	Receiver account type
Name	Receiver name
Identification	Receiver identification, corresponding to “Individual Identification Number” Nacha field
Data	Company discretionary data
Address	Address 1-3 of receiver
City	City of receiver
State	State of receiver
Country	Country of receiver
IAT Originating Bank. These fields are for international transactions only:	
Name	Name of the bank where the transaction originated
Identification	The originating bank identification
Country	The country code of the originating bank

Column	Description
IAT Receiving Bank. These fields are for international transactions only:	
Name	Name of the bank where the transaction is received
Identification	The receiving bank identification
IAT Correspondent Bank 1. These fields are for international transactions only:	
Name	Name of corespondent bank
Identification	The correspondent bank identification
Country	The country code for the correspondent bank

Payment details

Notes on payment holds display here.

Hold details

OFAC Scan: displays if hold has either been **Cleared** or **Escalated**.

Originate an ACH payment

1. Select the **ACH** tab.

2. Click **Originate Payment** to open the Originate Payment screen.

3. In the **Originate Payment** area, enter the payment details in the fields:

> 1. Aside from the fields described here, other fields may be displayed depending on your selection of the value for the **SEC Code** field.

Field	Description
SEC Code	SEC (Standard Entry Class) code for payment
Originator Account	Originator account number
Override Originator	Yes /No If Yes then **Originator Name**, **Originator Identification** and **Originator Data** display.
Receiver Routing #	Receiver account routing number
Receiver Account	Receiver account number
Receiver Name	Name on receiver account
Receiver Identification	Identification string receiver account.
Transaction Type	Transaction type (Push/Pull)

Field	Description
Service Level	Service type (Standard/ SameDay)
Description	Payment description
Purpose	Payment purpose
Client Identifier	Use this attribute to add your own unique identifying string to a payment call or COS record. This attribute is useful for idempotency purposes.
Amount	Payment amount
Addenda	Addenda for payment

4 Click **Originate Payment**. The payment details display in the ACH Payment Information screen.

Simulate inbound

Originations

1. Click the **Originations** tab to simulate an inbound origination and enter required information:

Field	Description
SEC Code	SEC (Standard Entry Class) code for payment
Originator Routing Number	Originator account number
Originator Name	Originator name
Originator Identification	Originator identification number
Originator Data	
Receiver Account	Receiver account number
Receiver Name	Name on receiver account
Receiver Identification	Receiver identification number
Transaction Type	Transaction type (Push/ Pull)
Service Level	Service type (Standard/ SameDay)
Description	Description
Amount	Payment amount

2. Click **Originate Payment**

4.5. Wires

Wires APIs

Cross River customer can transfer money electronically to a recipient, and a recipient can access the money, typically on the same day. Wire transfers are made through Fedwire®, operated by the US Federal Reserve.

What Cross River offers

- **Outbound wires**: Send a payment to an account outside Cross River
- **Inbound wires**: Receive a payment from an account outside Cross River
- **Fedwire**: Full support is available for wire transfers.
- **Domestic and international**: Account funding using wire transfers.
- **Drawdown request**: Request funds from an account outside Cross River to be sent to a Cross River account.
- **Drawdown response:** Respond to a request for funds received from an account outside Cross River.

Find out more

- Send and receive
- Payment types
- Payment authorization
- Times and availability
- Holds
- COS Explorer

Tutorials

See our tutorials to learn how to:

- Send a wire payment
- Simulate an inbound wire
- Send a drawdown request
- Respond to a drawdown request

4.5.1. IMPORTANT: Wires update

As part of the industry-wide transition to the ISO 20022 messaging standard, Fedwire will adopt this standard on July 14th, 2025. Migration to ISO 20022 enhances financial messaging, improves data quality, and increases efficiency across payment networks.

During Cross River's transition to the ISO 20022 standard we will maintaining existing functionality. The older messaging standard will be mapped to ISO 20022, with all changes occurring in the backend as primarily technical updates. **These backend changes will not impact partner-facing functionality**, ensuring that transaction initiation and processing remain consistent within our system.

Read more about it: **Fedwire Funds Service ISO 20022 Implementation Center | Federal Reserve Financial Services**

What you need to know

While COS functionality will largely remain unchanged, we are managing ISO 20022 compatibility in the backend to ensure a seamless transition with minimal impact to partners. While the message format will be updated, we will provide clear guidance to help ensure accurate data input and prevent errors.

We are introducing **new API validations and default values** due to the migration. We have updated the **Wires API documentation** to reflect these changes.

Here are a few examples of how transactions will be handled post-migration:

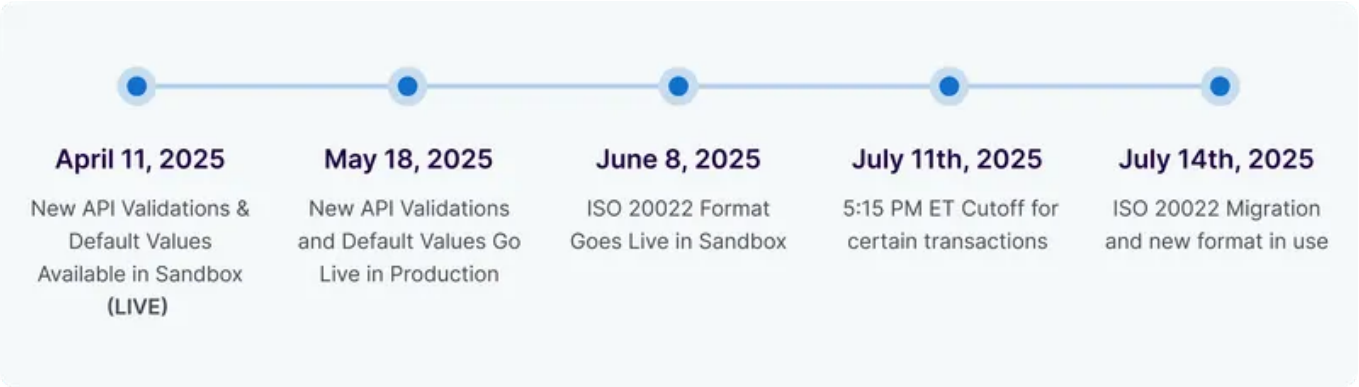
- CTRs (Customer Transfers) → Will continue to be processed as CTRs, though they will be converted to pacs.008 in the backend.
- SVCs (Service Transfers) → Will retain their current functionality with the following mappings:
    - SVC 1090 → Mapped to camt.110 (Investigation Requests) and camt.111 (Investigation Responses) in the backend.
    - SVC 1001 and 1007 → Mapped to camt.056 (Return Requests) in the backend.

- DRCs (Drawdown Requests) → Will still be processed as DRCs, though they will be converted to pain.013 in the backend.
- DRWs (Drawdown Payments) → Will remain DRWs, with responses to incoming drawdown requests handled as follows:
    - Negative responses → Mapped to pain.014 in the backend.
    - Positive responses → Mapped to pacs.008 in the backend.

We will also be making updates to the **Respond to Return Request feature**, which will take effect on July 14th, 2025**.** Summary of changes include:

- The **Request Service Message** button will be relabeled **Respond to Return Request**. This option will be used exclusively to respond to incoming return requests.
- You can initiate all returns (full and partial) using the **Reverse Wire** option. This provides a simplified and consistent way to process returns.

Change timeline and milestones



April 11, 2025 – New API Validations and Default Values Available in Sandbox (LIVE)

- These updates are now live in Sandbox. You can begin testing and adapting your systems in preparation for the upcoming migration.

May 18, 2025 – New API Validations and Default Values Go Live in Production

- These changes will be deployed in Production two months ahead of the ISO 20022 migration to give you time to validate in a live environment.

June 8, 2025 – ISO 20022 Format Goes Live in Sandbox

- The ISO 20022 message format will be enabled in Sandbox, allowing you to test your integration and gain familiarity ahead of the July 14, 2025 production launch.

July 11, 2025 – 5:15 PM ET Cutoff for certain transactions

- Customer Transfers (CTRs): Any CTRs initiated after 5:15 PM ET will not be processed on the same day and will be handled after migration on July 14th, 2025.
- DRCs and SVCs: We strongly advised you not to initiate the following transactions after 5:15 PM ET on July 11th to avoid potential processing delays:
    - DRCs (Drawdown Requests)
    - SVCs (Service Messages)
- If any transactions cannot be processed, your Relationship Manager will provide a list for re-initiation on July 14th, 2025.

July 14th, 2025 – ISO 20022 Migration and new format in use

- The ISO 20022 format officially replaces FAIM, and all payments will be processed using the new messaging standard.

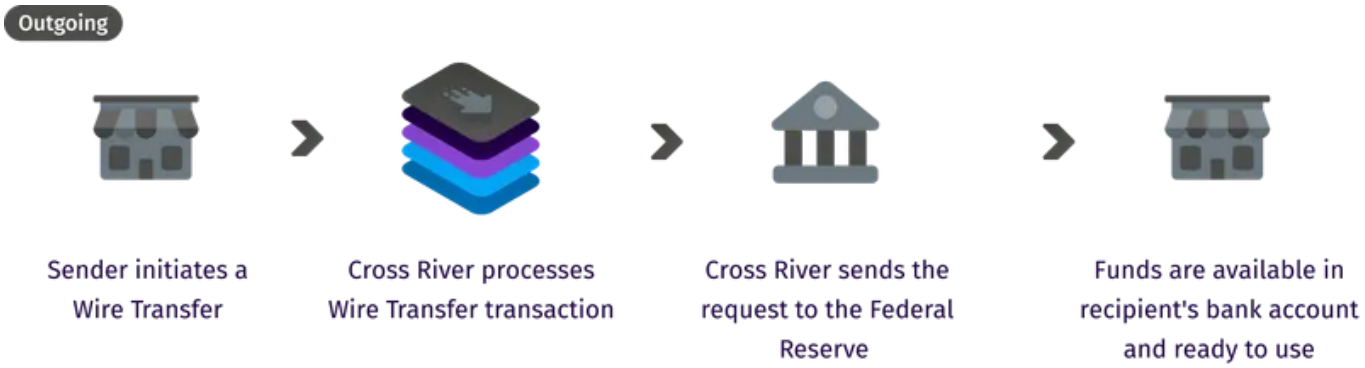
If you have any questions or need further assistance, reach out to your Relationship Manager.

4.5.2. Send and receive

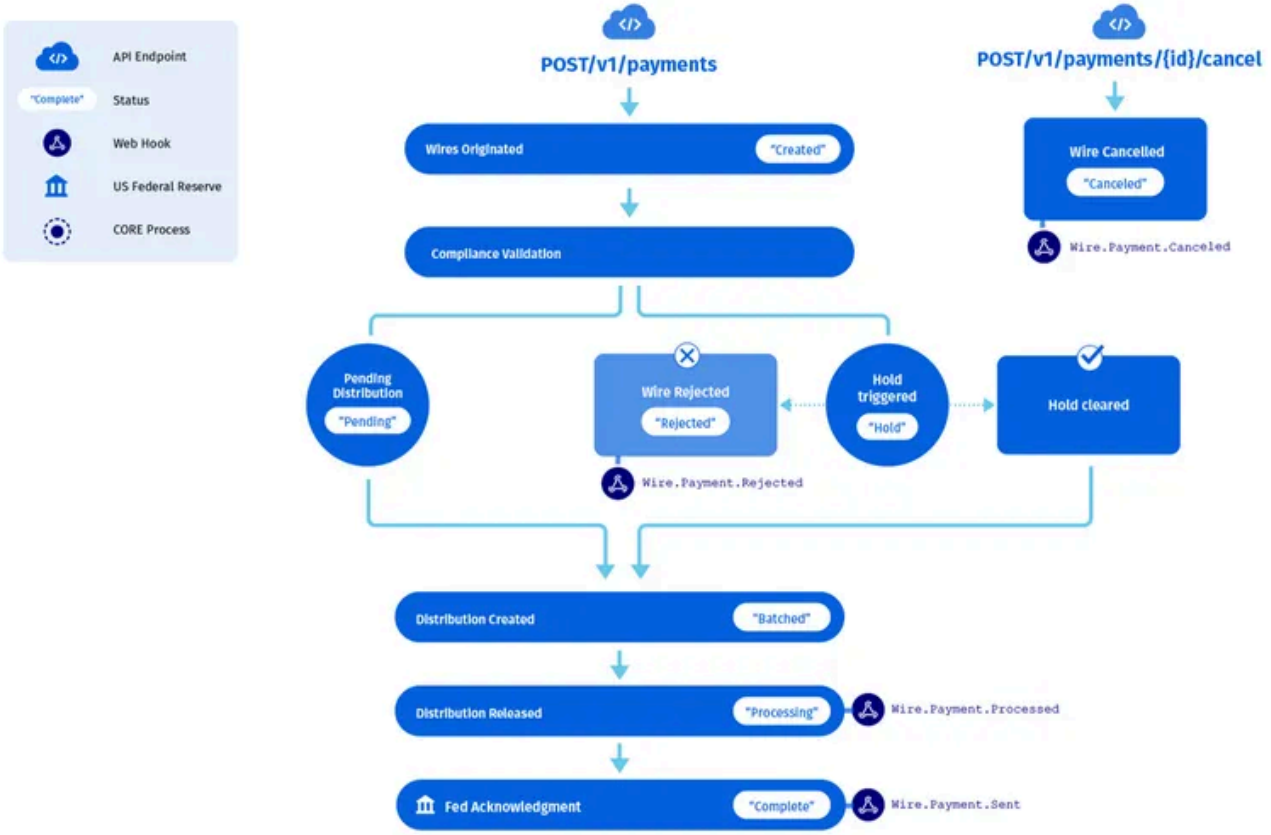
Outbound

When a customer uses **send an outbound wire** API, money is wired to an account at an external bank. When a wire payment is accepted and acknowledged by the Federal Reserve, an IMAD (IMAD) is assigned to the payment and returned in the API. This can be used to track the payment via the FedWire Funds Transfer system.

> See our tutorial to learn how to **send a wire payment**.

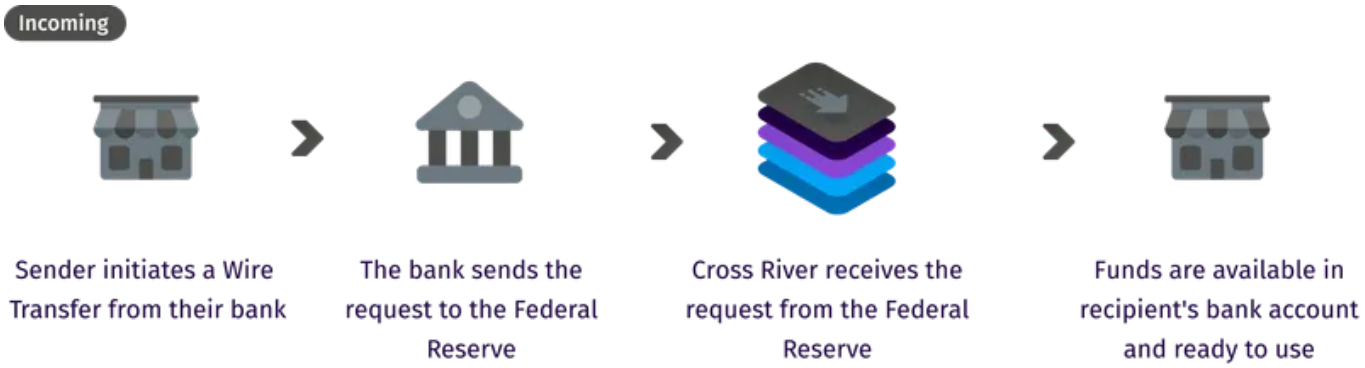


The following diagram shows the status and **webhook event** flows for outbound wire transfers:

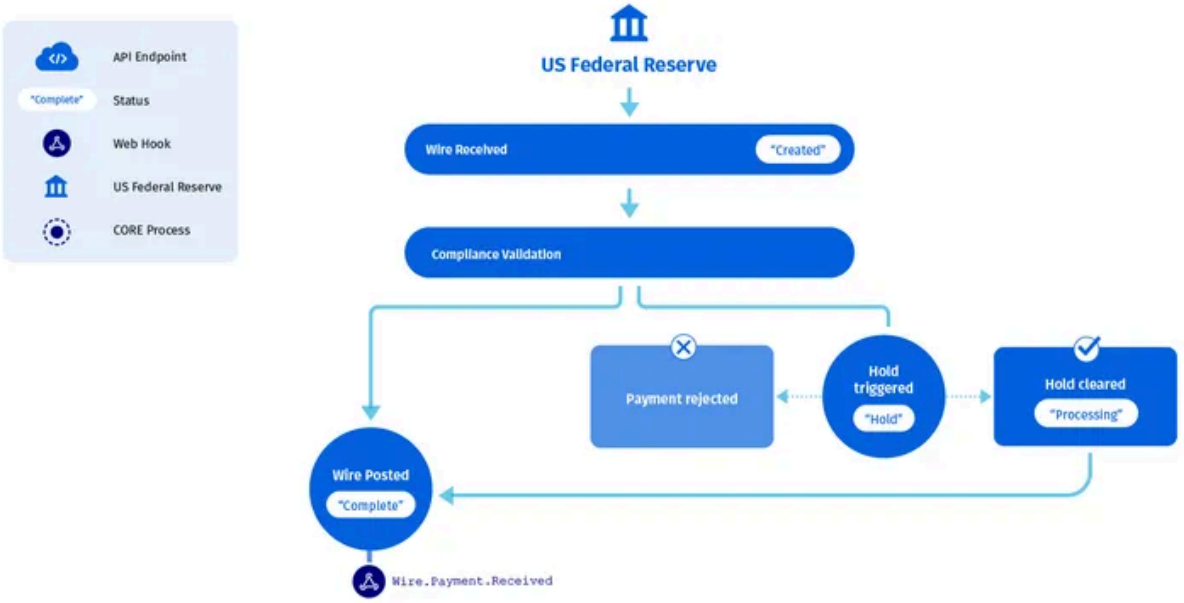


Inbound

When a customer receives an incoming wire, an external bank wires money to a Cross River account. You will receive a **webhook event** to indicate that a wire has been received.



The following diagram shows the status and **webhook event** flows for inbound wire transfers:


API Endpoint
"Complete"
Status
Web Hook
US Federal Reserve
CORE Process
US Federal Reserve
Wire Received
"Created"
Compliance Validation
Payment rejected
Hold triggered
"Hold"
Hold cleared
"Processing"
Wire Posted
"Complete"
Wire.Payment.Received

4.5.3. Payment types

A transfer is a new payment originating from either Cross River or another bank. Transfers can either be domestic or international. Both inbound and outbound international wire transfers in Cross River go through a domestic intermediary bank via Fedwire.

- **Domestic**
  Only US-domiciled banks are involved in the transfer. When a wire payment is accepted and acknowledged by the Federal Reserve, an Input Messaging Accountability Data (IMAD) is assigned to the payment. This can be used to track the payment via the FedWire Funds Transfer system.
- **International**
  An intermediary US-domiciled FI sends to or receives the funds from the Fed. Both inbound and outbound international wire transfers in Cross River go through a domestic intermediary bank via Fedwire. In addition, to send or receive international wires in USD, you *must* contact the beneficiary bank to obtain the appropriate wire transfer instructions.

Outbound wire transfer

1. Initiate the wire transfer via the **send a wire transfer** API or **originate a wire payment** in COS Explorer.
2. The wire transfer may or may not require authorization.
   - No authorization is required if the transfer is less than 1,000,000 USD, regardless of whether the transfer is initiated by API or in COS Explorer.
   - Authorize transfers in **COS Explorer** (2 approvers are required) if *both*:
     - The transfer is initiated by API
     - The transfer is between 1,000,000 and 5,000,000 USD dollars
   - The Cross River Wires Operation team calls the wire initiator if *either* of these cases apply:
     - The transfer was initiated using COS Explorer for more than 1,000,000 USD dollars
     - The transfer was initiated using the API for more than 5,000,000 USD
3. Once authorized, the wire transfer is sent.

> Only wires sent using the API can be approved by approvers set up in advance in COS Explorer.

Drawdown request

Request someone to send you funds using a drawdown request. This is basically a request for funds to be sent.

1. Initiate a drawdown request using one of these methods:
    - **Originate a drawdown request** API
      You can also follow our tutorial for **send a drawdown request**.
    - **Send a drawdown request** in COS Explorer
2. Provide your customer with the details below to send to the recipient of the drawdown request.
    - Cross River account number
    - Cross River routing number (**#021214891**)
    - Cross River bank address (**885 Teaneck Road, Teaneck, NJ 07666**)

Drawdown response

When Cross River receives a drawdown request, we respond by wiring the money that was requested to the bank account specified .

- See our **respond to a drawdown request** API
- See our **respond to a drawdown request** tutorial

Service Message

A non-value message sent from one bank to another for informational purposes only.

Reversal

A reversal of a previously sent wire transfer

4.5.4. Payment authorization

Payments over a certain amount have to be authorized, depending on the how the payment is **originated**. COS Explorer authorization is a feature that you can use instead of a manual callback by the Wire Ops team.

Callbacks

Under **certain conditions**, the Wire Ops team calls the wire initiator to confirm a request for a wire transfer.

- If the department is closed when the request arrives, the callback will be made the next business day.
- As a courtesy for our clients in the Pacific time zone, our team refrains from performing callbacks until 12:00pm EST.

Authorize outbound transfers

Authorizations (approvals) are only done through **COS Explorer** for *outbound* wire transfers originated via API.

- You can define either 1 or 2 approvers in the system in advance
- You can define 2 thresholds for approvals. The lower threshold would require 1 approver, and the higher would require 2. In any case, wires above 1,000,000 USD always require two approvers.
- You can configure the minimum thresholds that require authorization
- Use this feature to avoid callbacks from the Wire Ops team (for wire transfers above 1,000,000 USD but below 5,000,000 USD)
- It is a hold type in Explorer, designed to be cleared by you without needing someone from Cross River to clear the **holds**.

4.5.5. Times and availability

Times

- **Operating hours:** The Wire Operations Team is available 09:00 am EST to 06:00 pm EST, Monday through Friday, except holidays.
- Incoming wires received prior to 3:00pm EST are processed the same business day.
- Wire transfers are not processed on Federal Reserve holidays.
- For holidays that fall on a Saturday, Federal Reserve banks and branches are open the preceding Friday.
- For holidays that fall on a Sunday, Federal Reserve banks and branches are closed the following Monday.

Fund availability

- **Funds availability:** Funds from inbound and outbound transfers are available the moment they hit the recipient account if the transfer hasn't missed the cutoff time.
- Incoming wires received prior to 3:00pm EST are processed the same business day.
- Wire requests and payments made after your cutoff time will only become effective on the next business day. Ask your relationship manager for more details.
- Wire transactions are final once the funds are received.

4.5.6. Holds

A wire transaction on-hold status is a measure Cross River takes to protect our customers. Holds are a reflection of internal policies and regulatory requirements designed to mitigate risks like fraud, financial instability, or compliance violations. The **Wire Holds** area in COS Explorer displays information on the holds imposed on the wire, including the status of the hold and the type of the hold.

In some cases, system validations may trigger a hold for a variety of reasons explained below.

Type	Reason	Applies to	Clearing method
Beneficiary Account	The beneficiary account number in the payment did not match an account in COS	Inbound	Automated, Manual
Beneficiary Title	The beneficiary name in the payment did not match the beneficiary account title	Inbound	Automated, Manual
Catchy Scan	Some of the payment details potentially matched information within our fraud prevention database	Inbound, Outbound	Automated, Manual
Collected Funds	The core was unable to successfully collect funds for the payment from the originator's account	Outbound	Automated
Duplicate Suspect	Some of the payment details matched a payment that has been processed within the same business day	Inbound, Outbound	Automated, Manual
Inbound International	Some of the payment details indicate that the payment is an international transaction	Inbound	Automated, Manual
Internal Review	The payment was originated by an internal user via the COS Explorer portal	Outbound	Manual
Manual	Miscellaneous hold placed on a wire	Inbound, Outbound	Manual
OFAC Scan	Some of the payment details potentially matched information within the OFAC database	Inbound, Outbound	Automated, Manual
Outbound International	Some of the payment details indicate that the payment is an international transaction	Outbound	Automated
Authorization	The amount of the wire requires one or more approvals. One or more partner	Outbound	Manual

Type	Reason	Applies to	Clearing method
	users must access COS Explorer to clear the hold.		
Phone Confirmation	The payment was originated by an external user (partner) via the COS portal and must be processed using the Bank's callback verification procedure	Outbound	Automated, Manual
Travel Rule	The payment is missing some information that is required by the FinCen Travel Rule	Outbound	Automated

Wires must achieve a pending status to be eligible for distribution.

This happens when either:

- The validation process did not trigger any holds
- All triggered holds were manually cleared by a user

4.5.7. COS Explorer

Payments could include consumer payments, commercial payments, settlement payments with other networks or clearing houses, federal tax payments, or federal fund purchase and sale. You send, receive and settle a variety of payments on a same-day basis in the **Wires** tab.

When you set up your wires product, Cross River lets you define an allowlist of countries pre-approved to receive international wire payments from. International wires originating in countries not on your allowlist are checked by the Cross River Wires Ops team before approving the transfer. This may cause a delay in receiving the payment.

In the **Wires** tab, click **Wire Search** to display the **Wire Search** information screen.

View wire payment information

Wire details

Click an entry in the **Wire Search** list.

The main payment information displays in the **Wire Details** area:

Column	Description
Reference ID	Reference ID of wire payment
International	Whether the payment is international
Partner	Name of partner associated with payment
Status	Wire payment status
Payment Type	Wire payment type
Function Code	Business function code of wire payment
IMAD	IMAD (Input Messaging Accountability Data) number assigned to wire payment
Source	Wire payment source
Client Identifier	Optional field used for **idempotency handling**.
Created	Date and time wire was created
Auth	Whether authorization is Required/Not required
Completed	Date and time wire was completed
Amount	Amount of wire payment in dollars

Column	Description
Product	Product associated with account
Posting	Wire payment posting status
Direction	Wire payment direction
Type Code	Business type code of wire payment
OMAD	OMAD (Output Messaging Accountability Data) number assigned to wire payment
Purpose	Wire payment purpose
Effective	Date and time of wire payment effective
Dual Auth	Dual Authorization (Required/ Not required)
Sending Institution	Sending institution information
Originator Institution	Originating institution information
Originator	Originator information
Receiving Institution	Receiving institution information
Beneficiary	Beneficiary information
Sender Reference	Reference of entity sending wire

Posting records

The Posting Records dialog displays details of the wires attempts.

You can toggle between a display **All Attempts** or a display of only completed attempts.

Payment rasks

View Fedwire message

You can view the Fedwire message on the wire using the **View Fedwire Message** option in the **Payment Tasks** area.

In the **Fedwire Message View** dialog, the Payment ID displays with the Fedwire message itself.

View webhooks

The **View Webhook Events** dialog lets you filter on and see details about webhook events.

To filter for specific events, click the filter icon to toggle the search fields, and enter relevant search parameters. Click **Search**.

Column	Description
EVENT	Payment event
STATUS	Expired, Failed, Pending, Success
CREATED	Time and date stamp of event creation
ATTEMPTED	Time and date stamp of event attempt

Click Refresh button to resend a specific event.

Click OK button to open the **Webhook Event Details** dialog for a given event.

Click the entry under **RESOURCE** to display further details.

Click the entry under **CALLBACK URL** to view **Log Details**.

View history

The **Wire History** dialog displays the history of all related wire payments, including the direction of the wire, the function code, the date that the wire was received, the status and the reference ID number.

Click the bottom item in the **Payment History** area to view the Wire Payment Information screen for that related payment.

After clicking the bottom item, you can toggle the **Payment History** area to refresh the bottom row with the original payment you viewed. You can click the bottom row again to view the original payment again.

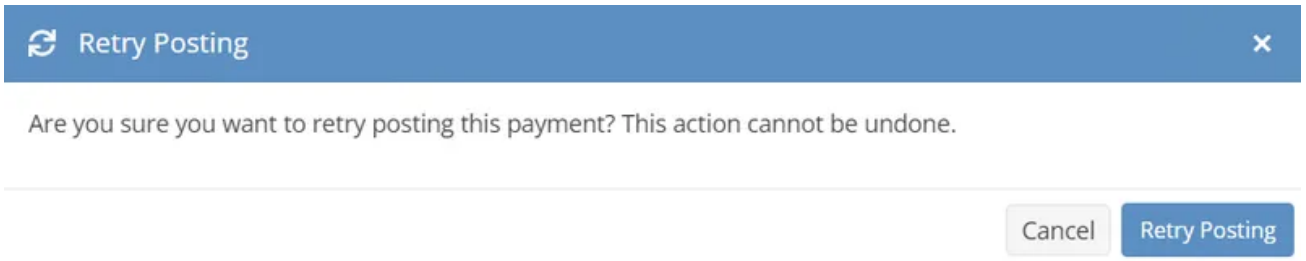
View notes

Click **View Notes** in the **Payment Tasks** area to view and add notes on a wire payment.

The **View Notes** dialog displays notes that have been saved for the wire payment:

Field	Description
DATE	Note creation date
SUBJECT	Note subject
AUDIENCE	Note audience (Internal /Partner)
CREATED BY	COS user who created note
Body	Note text

Click **Retry Posting**, this dialog displays:



Find out more

- Originate wire
- Originate drawdown
- Authorization
- Holds
- Find wire payments
- Return requests
- Payment templates

4.5.7.1. Originate wire

Steps

In the Wires module, the **Originate Wire** tab allows you to originate a wire payment.

To originate a wire payment:

1. Click the **Wires** tab.

2. Click **Originate Wire** to open the **Originate Wire** screen.

3. In the **Originate Wire** screen, enter the information into the fields:

Field	Description
Business Function Code	Business function code (**Required**)
Originator Account #	Originator account number
Receiver Routing #	Receiver routing number
Purpose	Purpose of wire payment
Amount	Dollar amount of wire payment
Beneficiary Identifier	Identifier type for the beneficiary
Beneficiary Name	Beneficiary name
Beneficiary Address [1-3]	Beneficiary address
Beneficiary FI Identifier	Identifier type for beneficiary financial institution (FI)
Beneficiary FI Name	Beneficiary financial institution name
Beneficiary FI Address [1-3]	Beneficiary financial institution address
Intermediary FI Identifier	Identifier type for intermediary financial institution
Intermediary FI Name	Intermediary financial institution name
Intermediary FI Address [1-3]	Intermediary financial institution address
Beneficiary Reference	Beneficiary reference code
Originator to Beneficiary Line [1-4]	Originator to beneficiary information

4 Click **Originate Wire**

The payment details display in the Wire Payment Information screen.

A window displays with confirmation that your drawdown request has been created.

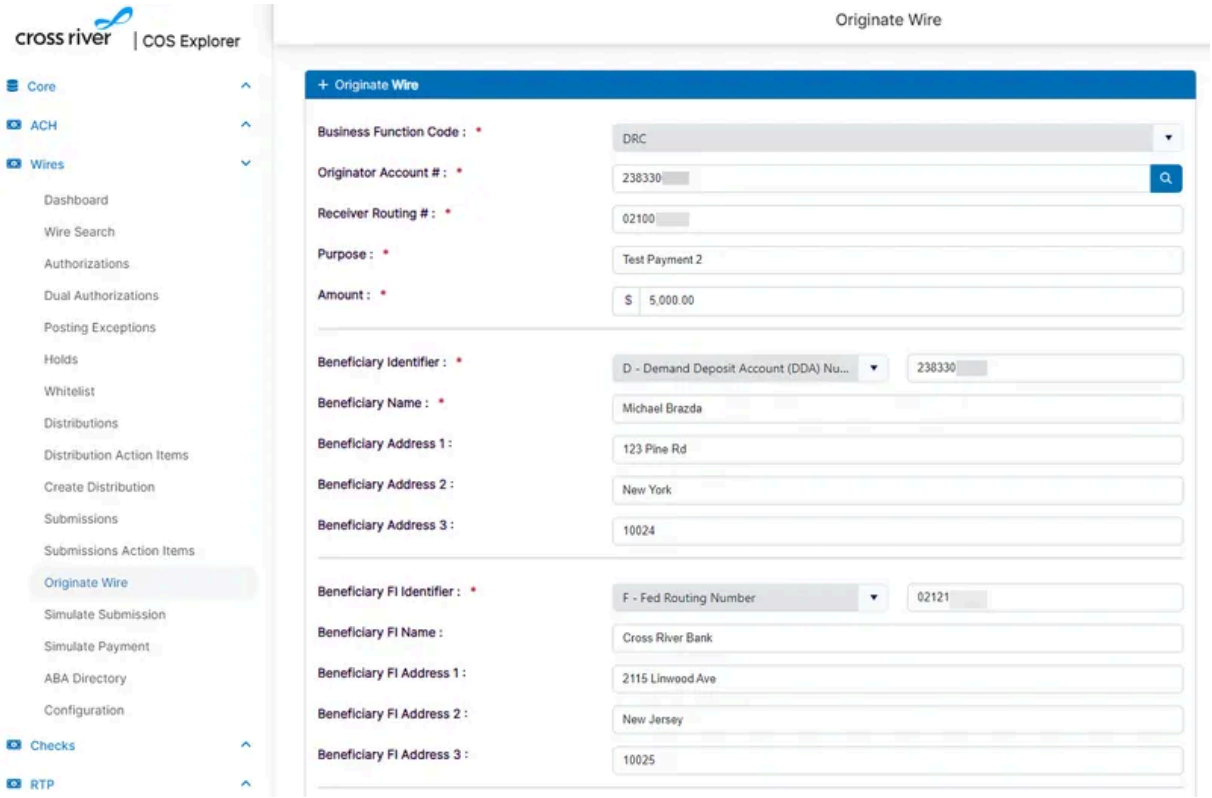
4.5.7.2. Originate drawdown

Steps

In the Wires module, the **Originate Wire** tab allows you to originate a wire payment. An outbound wire *drawdown* request, is also known as a reverse wire, involves requesting someone to send funds back to you, as the requester.

1. Click the **Wires** tab.

2. Click **Originate Wire** to open the **Originate Wire** payment screen.

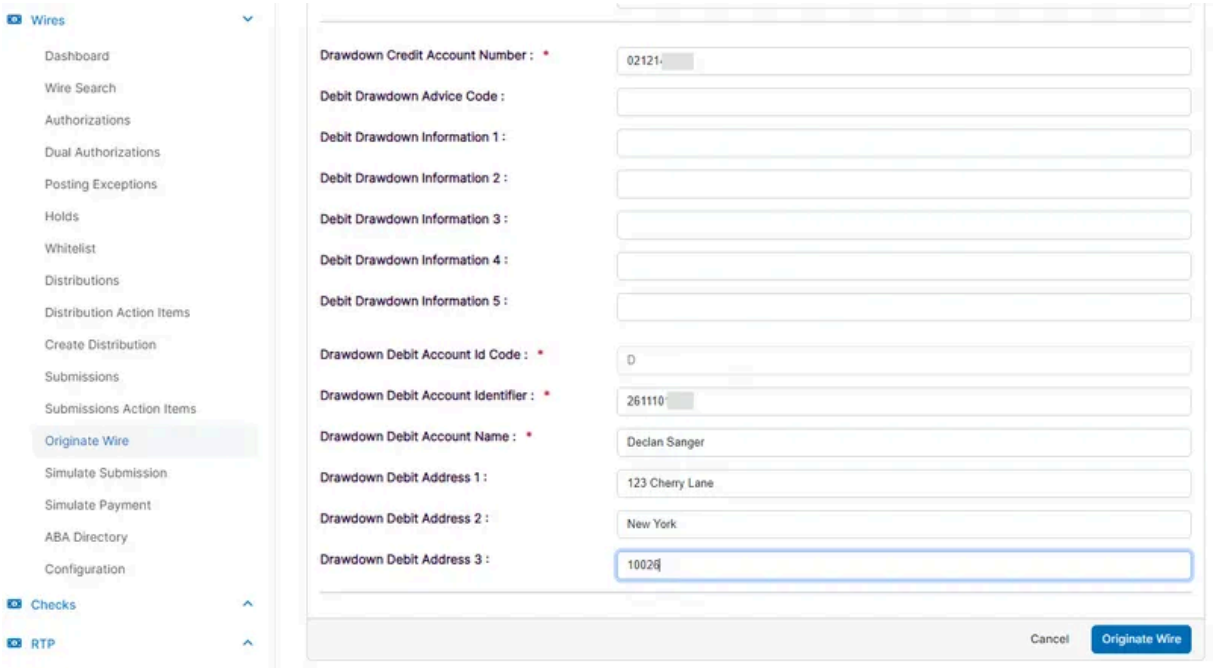


In the **Originate Wire** screen, enter the information into the fields:

Field	Description
Business Function Code	Business function code (**Required**). You must select **DRC**.
Originator Account #	Originator account number. This is your account #.
Receiver Routing #	Receiver routing number. The routing number where the drawdown request will be sent. For example, Chase or Bank America.
Purpose	Purpose of drawdown request.
Amount	Dollar amount of drawdown request.
Beneficiary Identifier	Identifier type for the beneficiary. You must select **D** and enter your account number as you will be receiving funds.
Beneficiary Name	Account holder (beneficiary) name
Beneficiary Address [1-3]	Account holder (beneficiary) address
Beneficiary FI Identifier	Identifier type for beneficiary financial institution (FI). Select F-Fed Routing number. Enter the 9-digit routing number for CR.

Field	Description
Beneficiary FI Name	Beneficiary financial institution name. Fill in the details of CR as we are sending the request on your behalf.
Beneficiary FI Address [1-3]	Beneficiary financial institution address. Fill in CR's address.
Drawdown Credit Account Number	Enter 9-digit CR routing number or 9-digit number of bank originating drawdown request.
Drawdown Debit Account Id Code	Enter D (Demand Deposit Account).
Drawdown Debit Account Number	The account number of the party who is being sent the request (whose account will be debited)
Drawdown Debit Account Name	The name on the Drawdown Debit Account Identifier.
Drawdown Debit Address 1-3	The full address of the person who receives the request for funds.

3 Click **Originate Wire**.



The payment details display in the Wire Payment information screen.

4.5.7.3. Authorization

IMPORTANT

DISCLAIMER:

This is a new process being provided as a service to customers of Cross River (CR). This process was created so that you can authorize wires online *yourself instead of receiving a callback* from Cross River.

As such, you may use this online wire authorization method instead of receiving a callback from Cross River, as long as your wire meets certain conditions mentioned below.

By opting to use this online authorization method *instead* of receiving a callback, you understand that:

- You now have more control of the timeliness of processing wires, but that the process does have additional risks.
- You are now authorizing these wires yourself, and Cross River will no longer verify the wire details with you before processing it.
- Once wires have been authorized by you, funds are directly released. There is *NO RECOURSE* once funds are released.
- You are responsible for:
  - Safekeeping credentials used by you, and your company to initiate, and authorize wires.
  - Setting up approvers in the system in advance to ensure you have enough approvers to provide dual authorization on wires.

See our **video demo**.

Verification and authorization

To protect you and your customers, Cross River may require you to verify the details of your wire before it can be processed. Traditionally, we only obtained this verification using

a callback, where we would call you or another member of your team to verify details of the wire.

Authorization is a feature that you can use instead of a callback to verify the details of your wire using COS Explorer. It is a hold type in Explorer, designed to be cleared by you without needing someone from Cross River to clear the hold. Read more about **holds**.

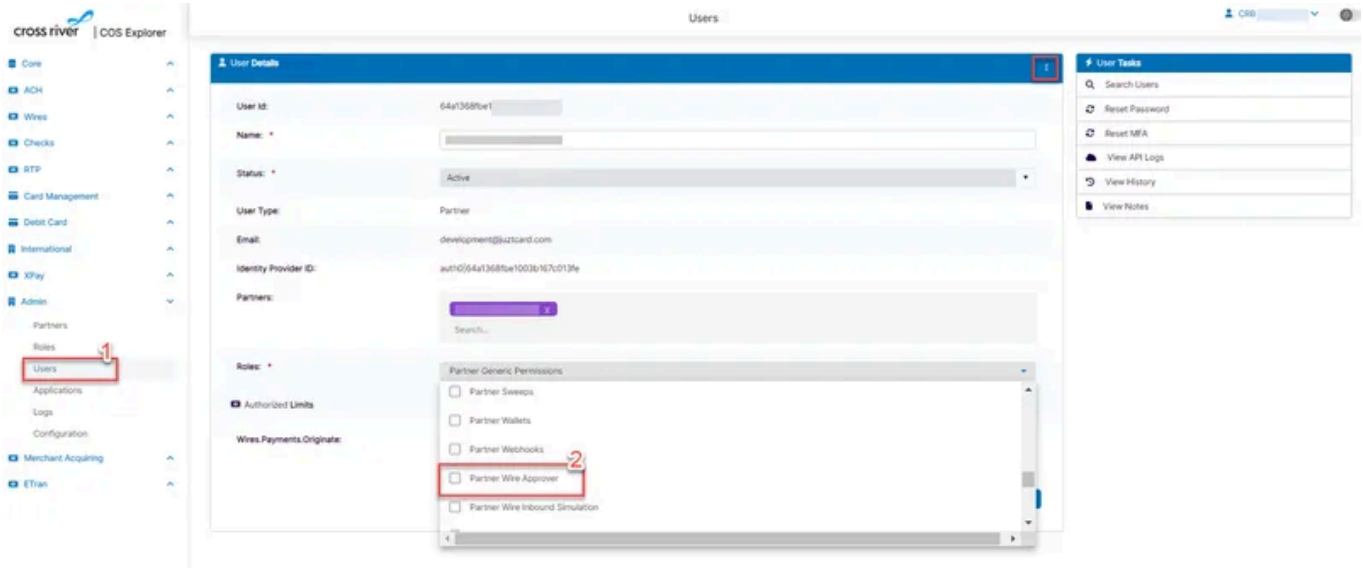
Some important details about Authorization:

- Authorization can be configured at the product level or overridden at the account level.
- Cross River only allows Authorization as an alternative to callbacks for wires up to a certain amount. If we require a callback, your wire will be placed in a **Phone Confirmation** hold. This is how the current process works.
- For Authorization to be used instead of a callback, Cross River requires dual authorization, meaning 2 different people must approve the wire for the hold to clear in Explorer. In addition, the originator of the wire cannot be one of those 2 approvers. So you may need up to 3 different people in total.
- You can still use Authorization as your own "maker-checker" workflow even when Cross River doesn't require any verification (callback or authorization) to process your wire (which would only apply if the wire is less than a certain amount).
  In this case, you are free to customize thresholds in the system up to a certain amount so that you can require either single or dual authorization.
- In addition to the amount, depending on the source of your wire (whether originated through your own application via an API call to COS or through Explorer directly), Cross River may still need to place a callback to you before your wire can be processed.
- Cross River will never require you to authorize your wires using both authorization methods (both callback and via COS Explorer). However, if you'd like to authorize your wire using both authorization methods, that is up to you.

Partner wire approval role

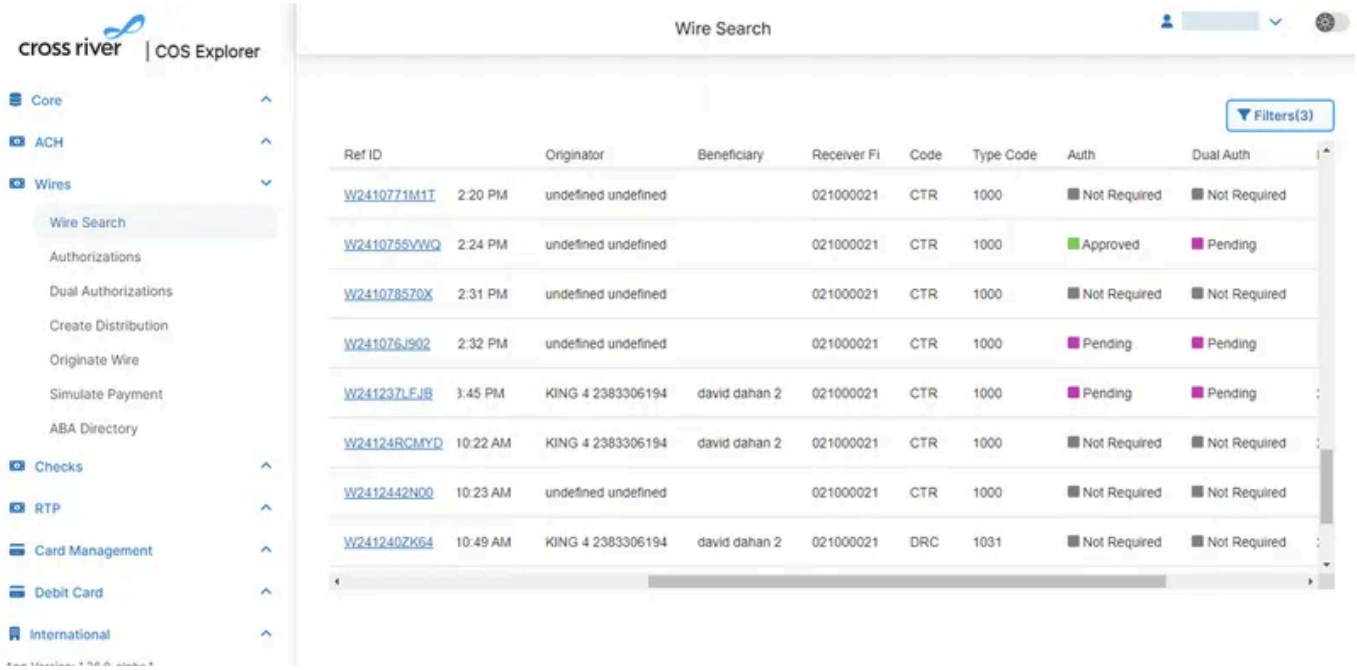
To authorize wires in COS Explorer, you must first assign users the **Partner Wire Approver** role. Only an admin user from your team can perform this task, and admins are designated during your onboarding and account setup.

1. In the **Admin** menu. Click **Users**. The **User Details** screen displays.
2. Click the 3-dot menu to edit the fields and select **Partner Wire Approver**. Please note that there is only one **Wire Approver** role and it can be used to clear both **Auth** and **Dual Auth**.



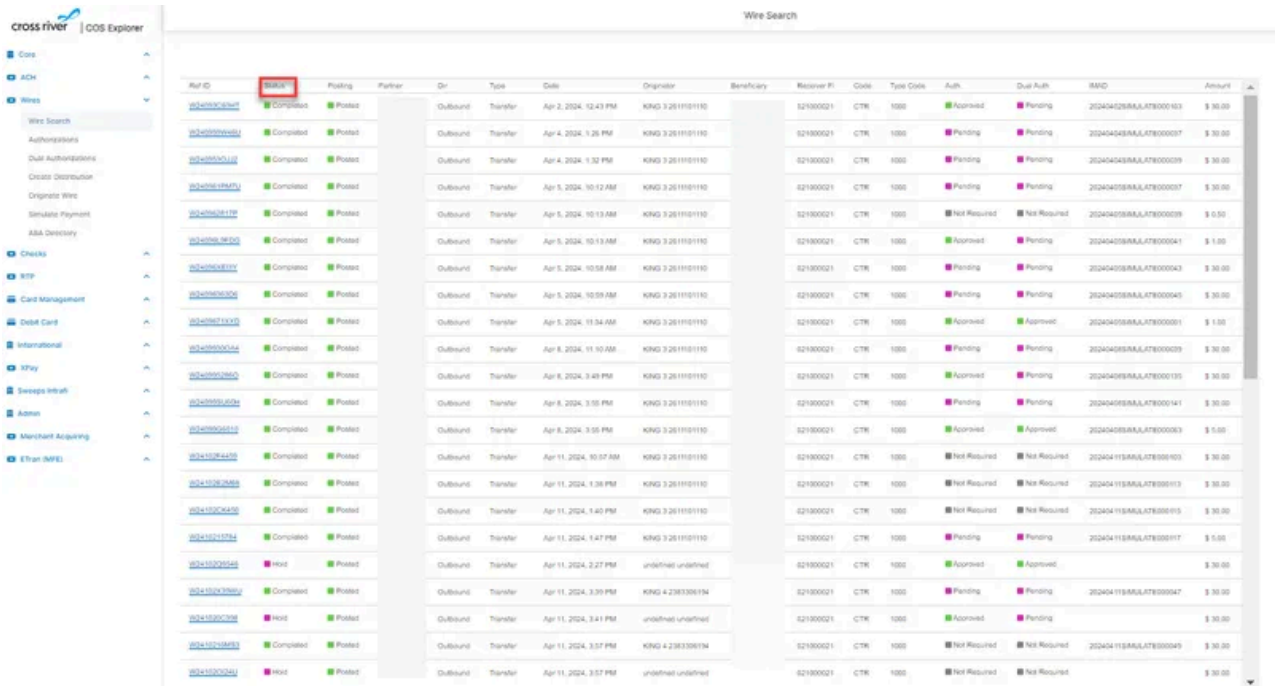
Authorize a wire

When a wire is originated above your verification threshold amount it goes into **Pending** status.



The Auth and Dual Auth columns display statuses with colored boxes. The color of the box indicates the action required:

1 Find your wire so you can approve it. Under **Wires**, click **Wires Search** and look for wires that have a **Pending** status.

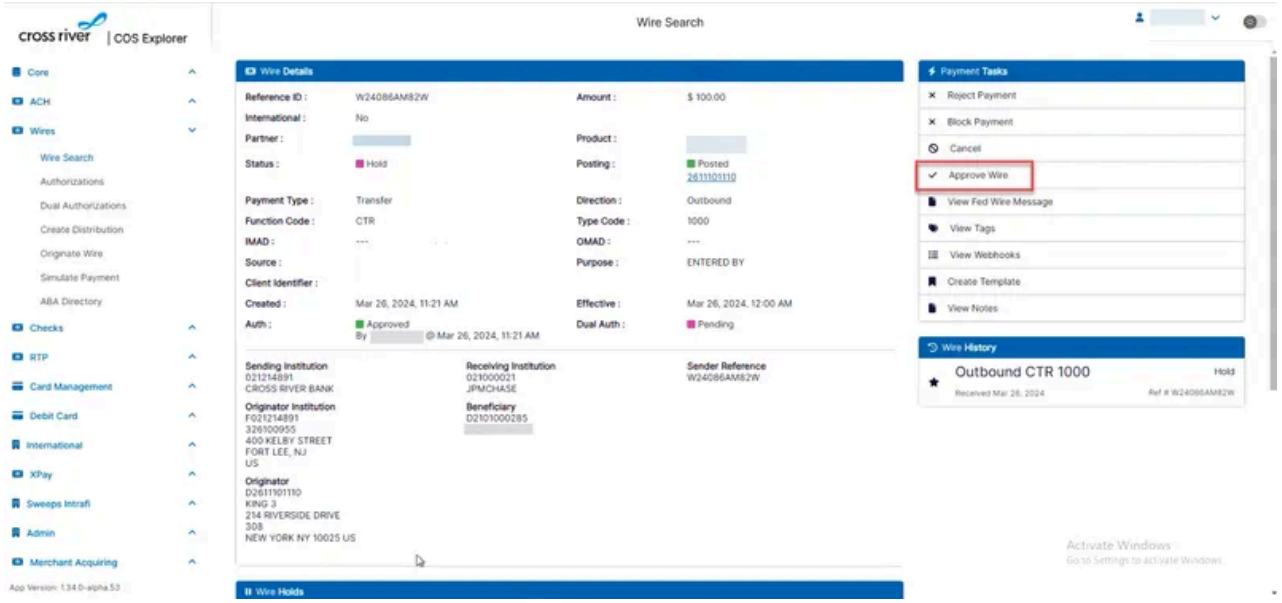


The **Auth** and **Dual Auth** columns display statuses with colored boxes. The color of the box indicates the action required:

- A pink box with **Pending** status under either column means the wire must be verified via Authorization before it is sent for processing. If there is a pink box with **Pending** status under both columns, then 2 different users will have to approve.
- A green box showing **Approved** means that approval was provided for the respective **Auth**.
- A grey box showing **Not Required** means the wire does not need to be verified via Authorization.
- If only the **Auth** column contains a pink box showing **Pending** , only one user is required to clear the Authorization.

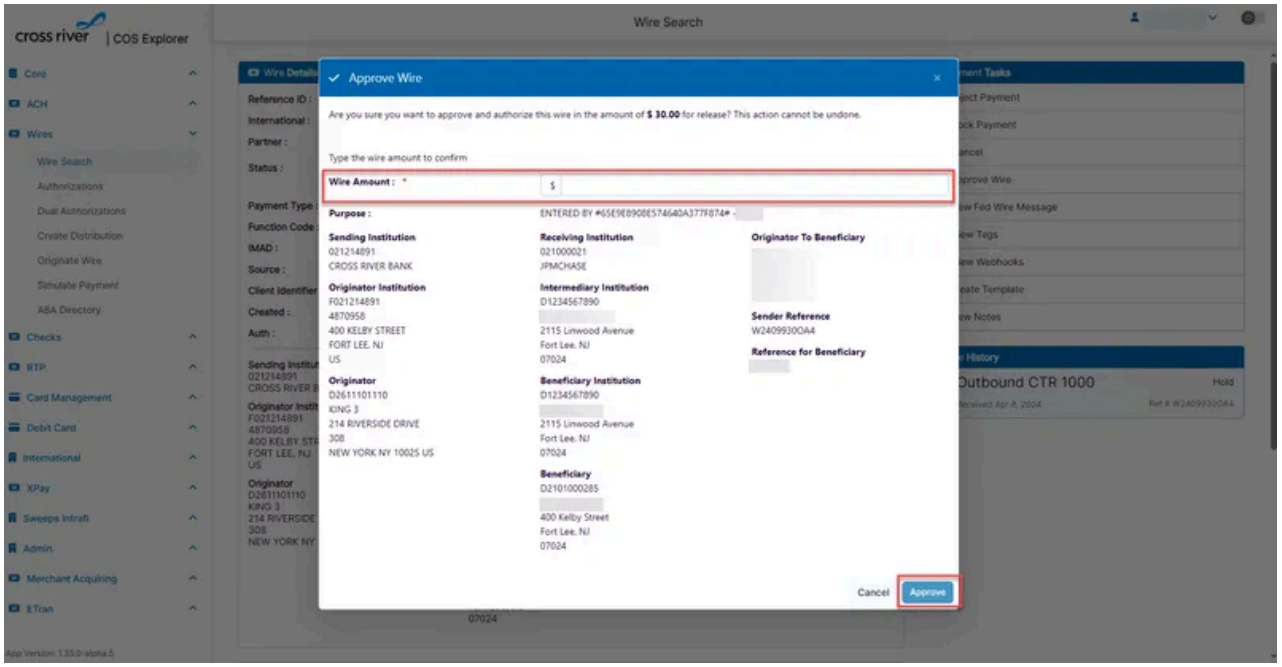
2 Click the wire that needs the review. The **Wire Details** screen displays.

3 If the wire is ready for authorization, go to **Payment Tasks** and click **Approve Wire**. *There is no expiration on wires that require Auth or Dual Auth.*



> Make sure you have enough users with the correct access to approve wires in these holds. The user who originates the wire and the wire approvers must be different users, and may require up to 3 people total.

4 The **Approve Wire** pane displays. Enter the **Wire Amount** and at the bottom of the screen, and click **Approve**. Make sure you review the wire details before you approve it.

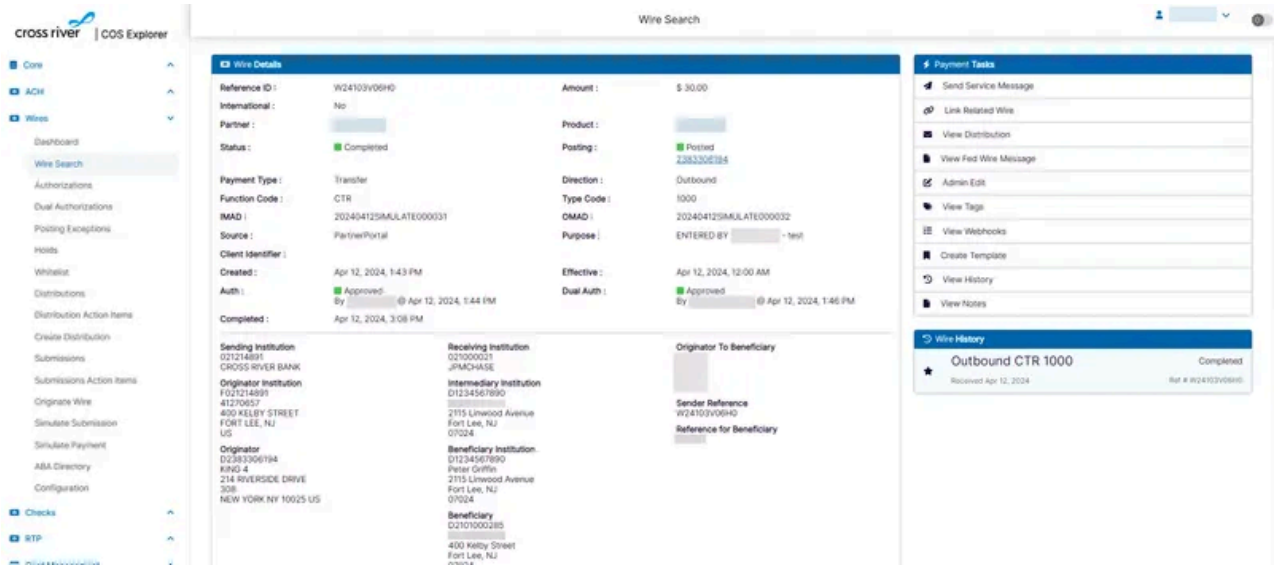


5 When the first approver approves the wire, **Auth** displays in the **Wire Details** screen with a green box indicating the wire was approved, along with who approved it and

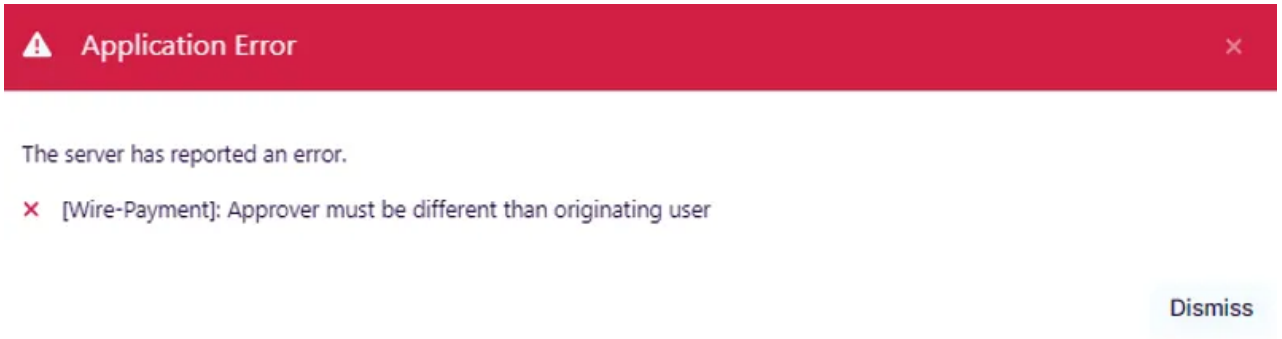
when.

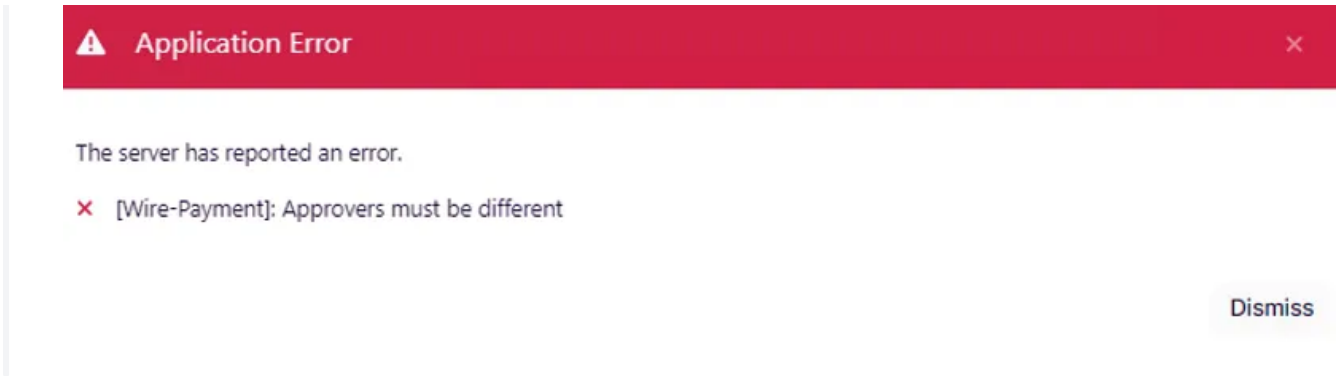
If the wire requires **Dual Auth** approval, then when the second approver completes the approval, **Dual Auth** displays with a green box as **Approved**, along with who approved it and when.

The approval process for both **Auth** and **Dual Auth** are the same. If the wire requires dual authorization because it exceeded both your **Authorization** and **Authorization 2** thresholds, then once the first user approves **Auth**, a different user must follow steps 1-5 above to approve **Dual Auth**.



If the same user who just approved **Auth** or the wire originator attempts to approve the wire, an error message displays.





You will **not be notified** that it's your turn to approve a wire that's in an **Authorization Hold**. It is your responsibility to ensure you monitor the wires that require approval. After the **Authorization Hold** clears, the wire may also be subject to other holds during the screening process.

4.5.7.4. Find wire payments

Steps

1. Go to **Wires**>**Wire Search**.

2. **Search by ID** or click the filter icon and enter your search parameters:

Filter	Description
Ref ID	Wire payment Ref ID
Partner	Partner name
Product	Product name
Auth	Not Required, Pending, Approved, Canceled
Dual Auth	Not Required, Pending, Approved, Canceled
Direction	Payment direction (Inbound /Outbound)
Type	Wire payment type
Status	Payment status
Start	Payment date, search from that date onwards
End	Payment date, to search until that date
Min Amount	Minimum payment amount
Max Amount	Maximum payment amount
Imad	IMAD (Input Messaging Accountability Data) number
Ben. Name	Beneficiary name
Orig. Name	Originator name
Reversed	Whether the payment has been reversed
International	Whether the payment is international
Business Func	Business function
Type Code	Transaction type code
Client Identifier	Use this attribute to add your own unique identifying string to a payment call or COS record. This attribute is useful for **idempotency** handling.

Filter	Description
Blocked Reason	Reason for wire being blocked
Rejection Reason	Reason for rejection
Source	API, Partner Portal, Ops Portal

3 Click **Search**.

Each row in the list represents an individual wire payment.

The search results display in the **Wire Search** list:

Column	Description
Ref ID	Wire payment Ref ID
Status	Wire payment status
Posting	Wire payment posting status
Partner	Partner associated with wire payment
Dir	Direction of wire payment
Type	Wire payment type
Date	Wire payment date
Originator	Name and account number of payment originator
Beneficiary	Payment beneficiary name
Receiver Fi	Receiver routing number
Code	Business function code
Type Code	Code describing transaction
Auth	Authorization required/not required
Dual Auth	Dual authorization required/not required
IMAD	IMAD number of wire payment
Amount	Amount of transaction

4.5.7.5. Return requests

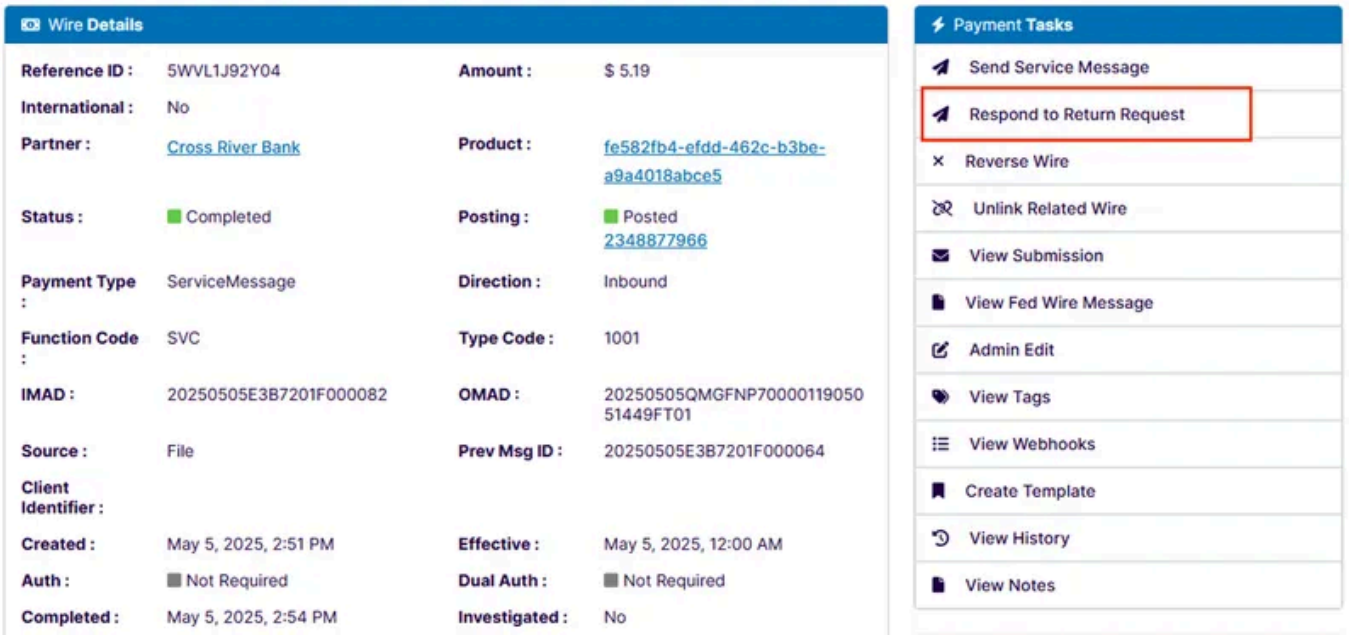
Cross River has introduced the **Respond to Return Request** feature in COS Explorer for wire transactions, designed to improve efficiency in managing wire returns. This new feature allows you to indicate whether you will return the full amount, a partial amount, or no amount for a given wire transaction.

The **Respond to Return Request** feature includes a convenient drop-down menu for your responses, streamlining the review process and expediting the release of service messages.

Use this feature instead of the Notes section.

For questions, contact your **Relationship Manager.**

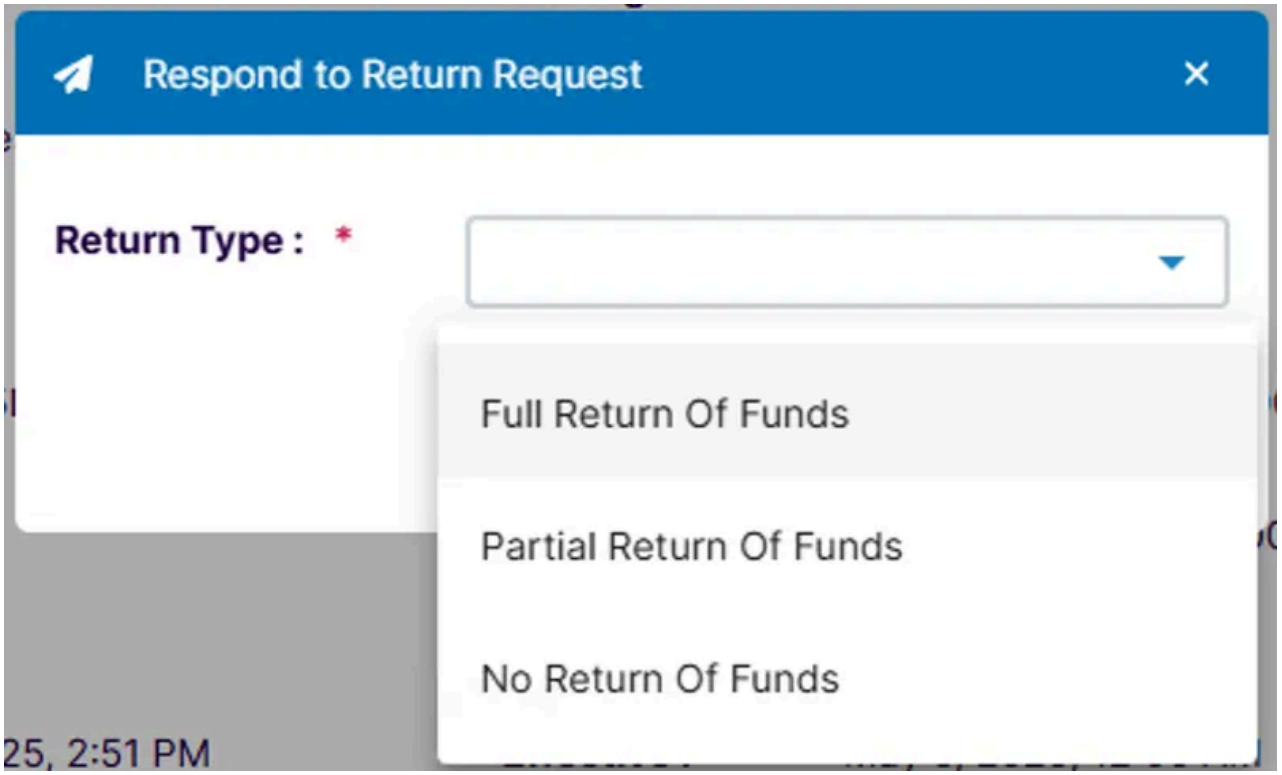
For an incoming return request, click **Respond to Return Request** in **COS Explorer** from the list of payment tasks in the right-hand pane.



Initiate a return response

1 **Select an incoming return request**

1. Navigate to the relevant **wire transaction**, typically an inbound wire for which a return has been requested.
2. Click **Respond to Return Request**. A drop-down menu appears.



2 **Choose an option.**

Each option has specific input requirements.

- **Full Return of Funds**
  Select this option if the full amount can be returned. **You must initiate the payment return using the "Reverse Wire" functionality after responding to the return request.**
- **Partial Return of Funds**
  Enter the partial amount that will be returned. **You must initiate the partial payment return using the "Reverse Wire" functionality after responding to the return request.**
- **No Return of Funds**
  Select one of the following reasons from the drop-down menu:

Reason code	Reason type	Description
AC04	Account closed	The account number specified is closed on the Receiver's books
AM04	Insufficient funds in the account	There are not enough funds available to cover the specified message amount
ARDT	Funds already returned	The return request isn't accepted because the transaction was already returned
CUST	Customer declined the return	The return request cannot be accepted because of a customer decision (Creditor)
IDMN	Hold Harmless	The return request cannot be accepted until indemnification is received
Other	Answer using free text	Use only when the other options don't apply

4.5.7.6. Payment templates

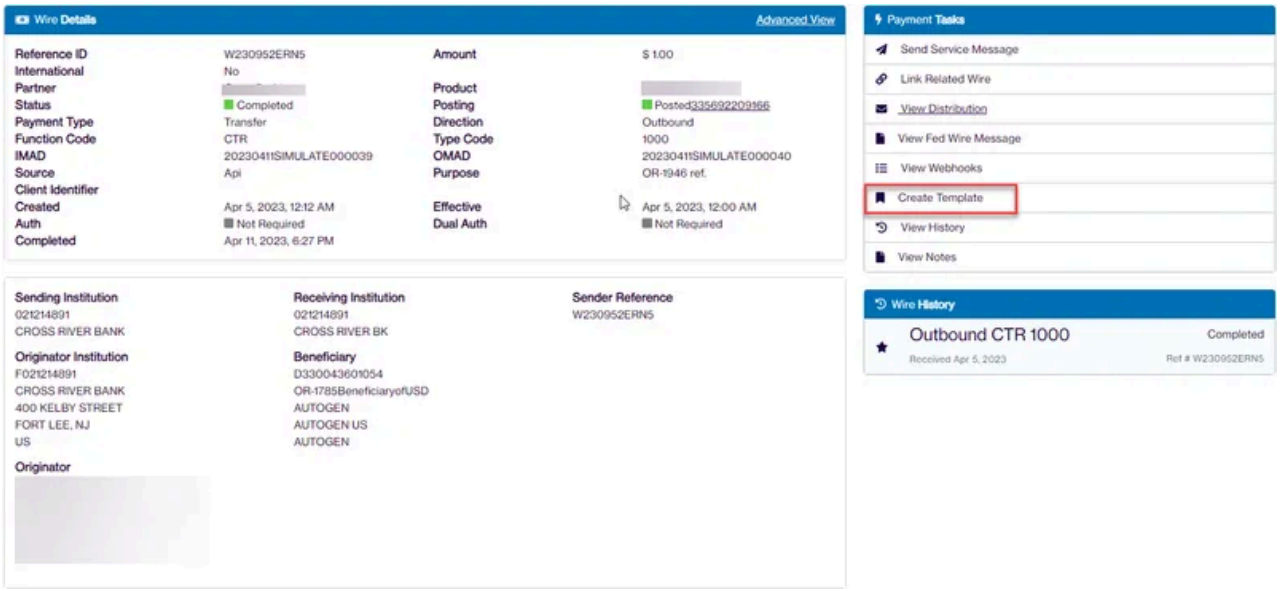
A payment template lets you pre-define the parameters of a recurring payment with the same or similar details. This lowers the need for daily manual payment set up.

> A payment template must be based on an existing payment in COS Explorer.

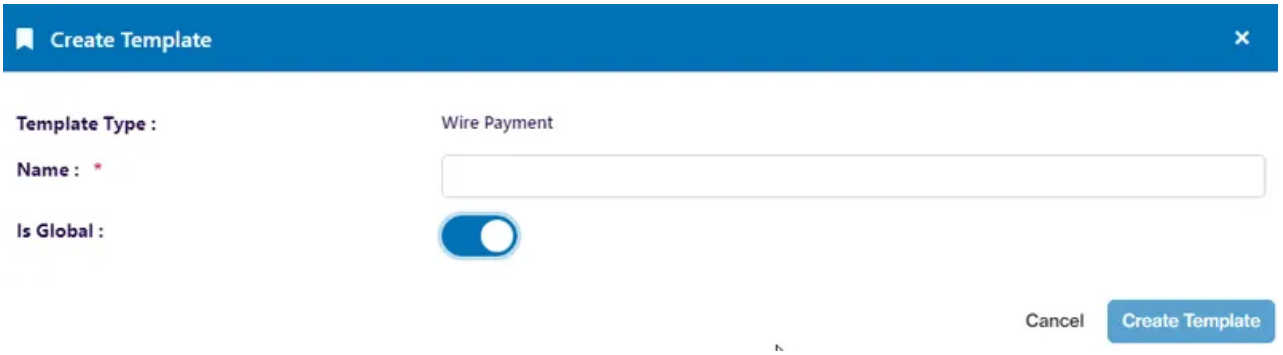
Create a template

COS Explorer supports payment templates for all payment rails. This example shows a Wires payment template.

1. Navigate to the **Payment Details** page of the wire you wish to use to create a payment template.
2. Click **Create Template.**



3. **Create Template** dialog displays.



4. Name your template. Template names must be unique and cannot be duplicated.

5. Toggle **Is Global** if you want to make the template visible to others.

6. Click **Create Template**.

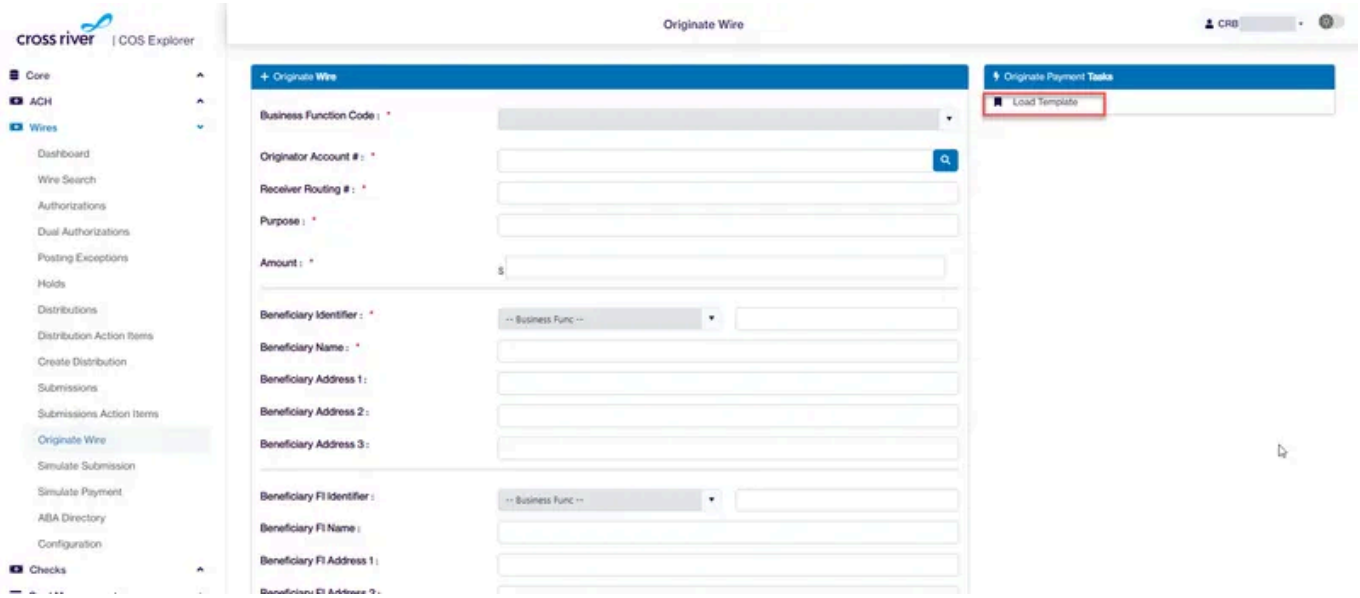
Access your template via the **Originate Wire** screen

> This template contains all details for a future payment, including the amount.

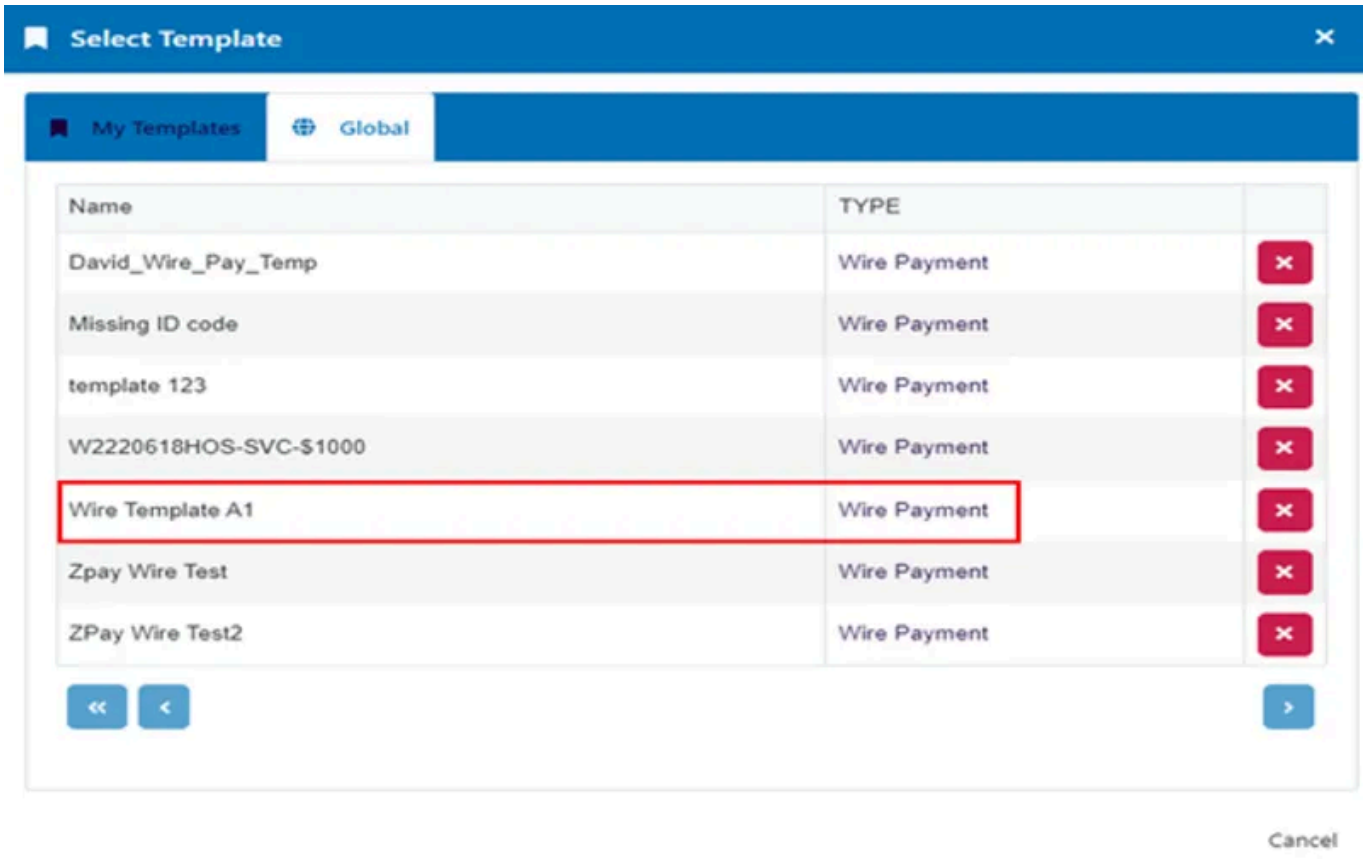
Use a template

1.To access a template, navigate to **Originate Wire**.

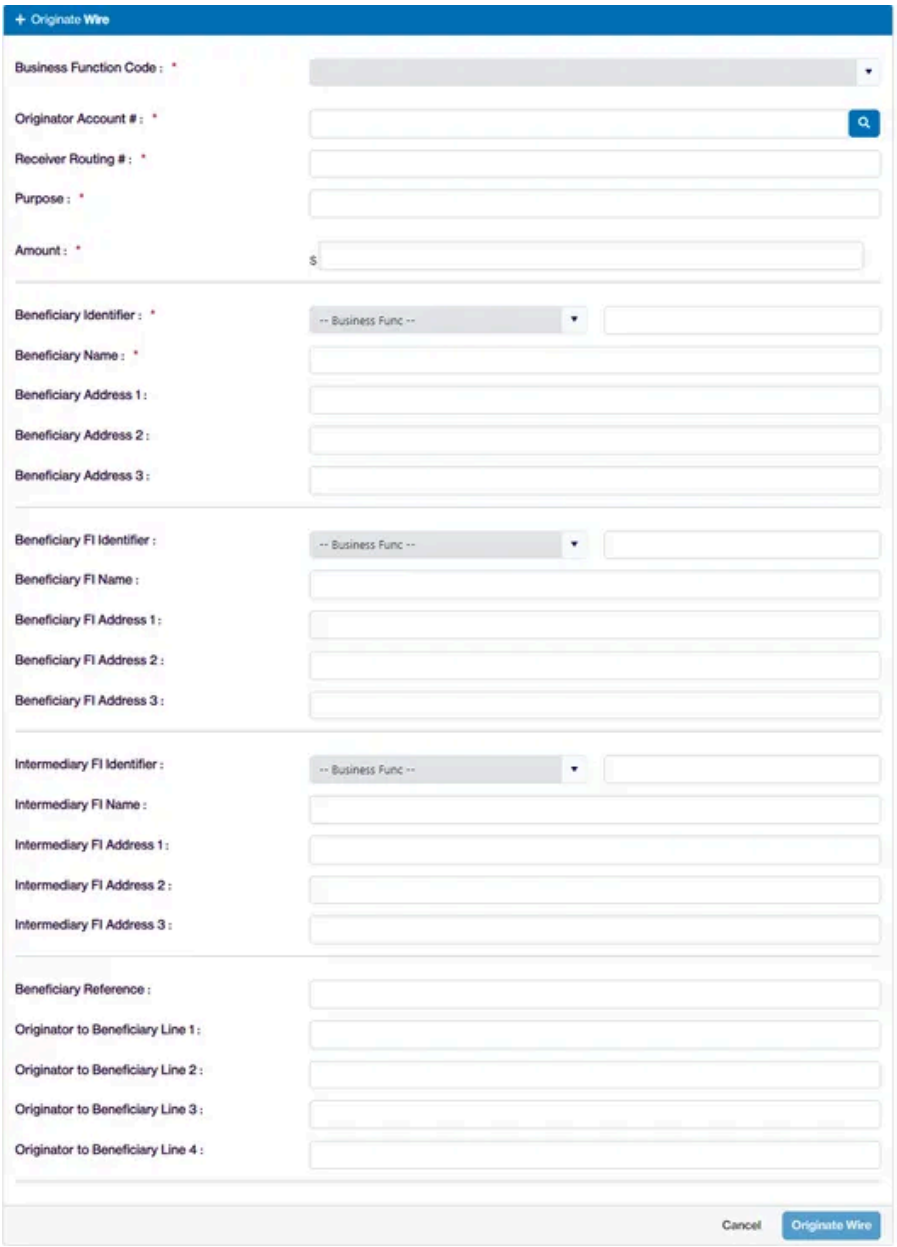
2. In **Originate Payment Tasks** pane click **Load Template:**



3. The **Select Template** page displays. Select your template:



4. The **Originate Wire** page displays with the payment template details filled in:



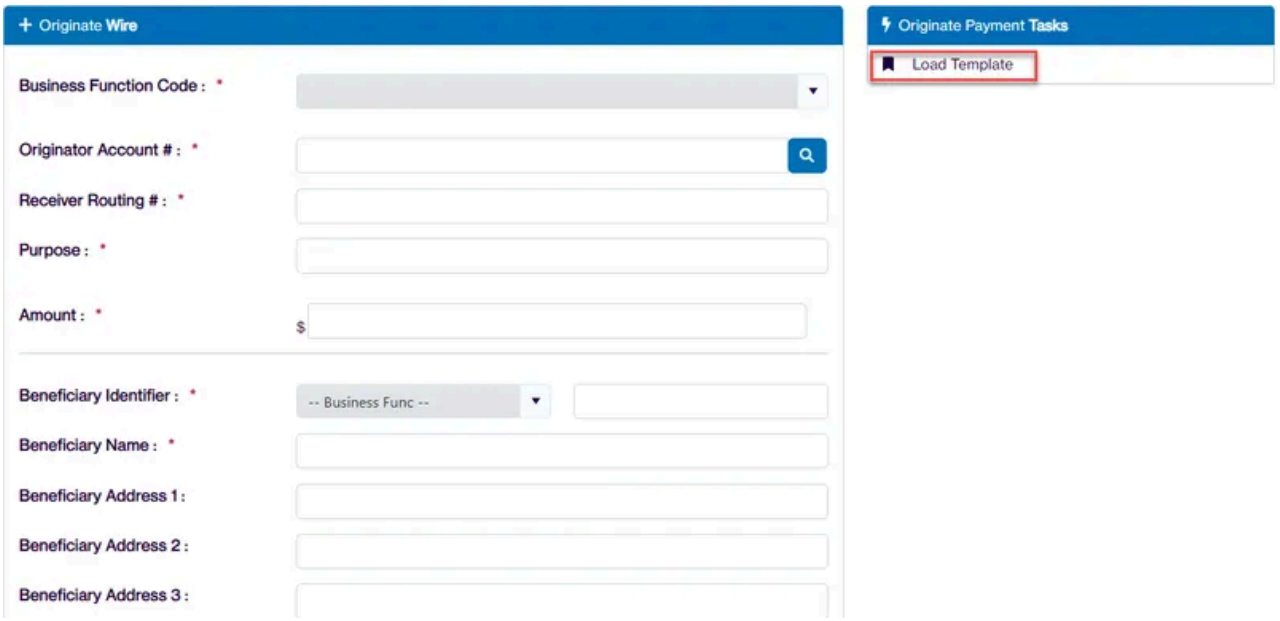
5. Review the details and change any information, if necessary, especially payment amount.

6. When everything is correct, Click **Originate Wire**.

7. View your wire on the **Wire Details** screen.

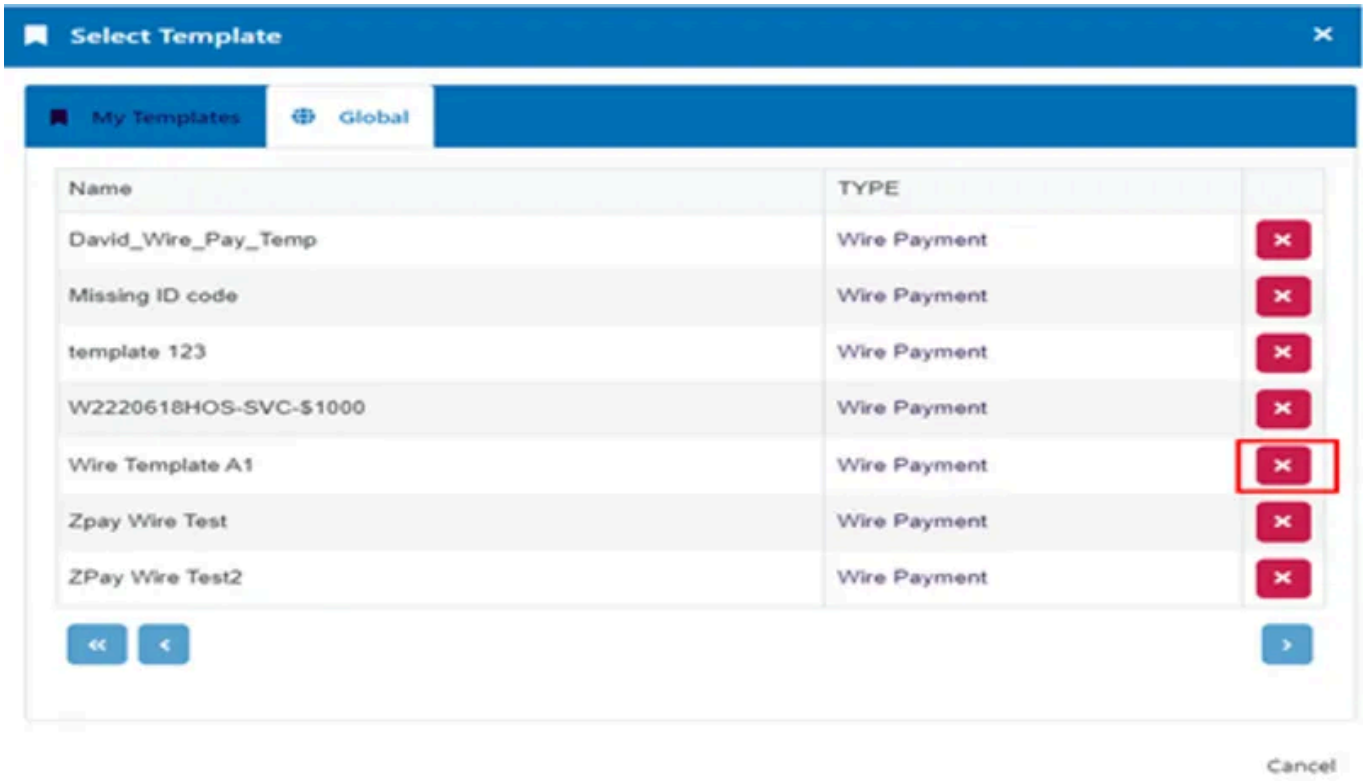
Delete a template

1. Navigate to **Originate Wire** and click Load Template



Select Template displays.

1. Find the template you want to delete.
2. Click on the X button to remove the template.



4.6. XML batch payments

Cross River's XML batch payment (ISO20022) channel provides you with a way to send all your instructions in one single file format (XML).

The system:

1. Processes input files that you have supplied.
2. Records financial movements.
3. Sends Payment Status Reports (PSRs) as output files to update you about payment processing, rejections, and returns.

The input files must conform to the ISO 20022 PAIN.001.001.03 and PAIN.008.001.02 standards, and the PSRs conform to the ISO 20022 PAIN002.001.03 standard.

What Cross River offers

Cross River supports these payment methods (rails):

- **Book transfer**
- **ACH CCD/PPD**
- **Fedwire**
- **ACH direct debit**
- **RTP/FedNow**

The rail that you use for payment processing depend on your agreement with Cross River. The client support team sets up the rails during your onboarding process.

You will learn how to create the files correctly and how to send them to Cross River. We've included **examples** of input files and PSRs, together with explanations of their blocks and elements. We've also included **return statuses and error codes** so you can understand what's happening with your files.

> You can find detailed information about the ISO 20022 message definitions on the **ISO site**.

You can reach out to **support** with any questions.

Acknowledgment files containing initial processing information are normally available within minutes. After that, Cross River checks for updates based on rail type.

Find out more

- **Payment methods**
- **File transfer**
- **Requirements per rail**
- **Prepare input XML file**
- **Output files**
- **Validation**

4.6.1. Payment methods

XML batch payments (ISO 20022) promotes efficiency, accuracy, and interoperability across payment networks, making it essential for anyone involved in financial transactions.

Direct debit (ACH)

Cross River's **ISO 20022 pain.008** implementation supports creation of direct debit instructions using the US ACH system.

Cross River checks for returns and NOCs once on the effective date provided in the response, then once a day for an additional 3 business days, and then once a week for up to 6 months.

Corss River checks for `Processed`, `Rejected` and `OnHold` statuses once on the effective date provided in the response, and then once a day for an additional three business days.

The checks happen at 20:00 EST for ACH.

> Effective date is the day the payment came in, or the next business day if the payment came in after the cutoff time set in COS.

ACH push (CCD/PDD)

Cross River's **ISO 20022 pain.001** implementation enables you to initiate electronic payments (such as payroll or vendor payments) through the ACH network.

Fedwire

Cross River's **ISO 20022 pain.001** implementation supports Fedwire payments.

Cross River checks for status updates at 18:00 EST on the business day that the payment was supposed to be processed, and also at 18:00 EST on the business day after that.

Book transfer

Cross River's **ISO 20022 pain.001** implementation allows you to send book transfers to any bank account

Cross River checks for status updates every 30 minutes.

RTP (real-time payments)

Cross River's **ISO 20022 pain.001** implementation supports real-time fund transfers from The Clearing House (RTP) and from the Fed (FedNow).

4.6.2. File transfer

We use SFTP to enable customers to upload their payment instruction files and receive the reports. Each customer will receive a unique SFTP location for upload and download.

To access the SFTP

1. **Set up SFTP**.
2. Access your SFTP folder using the credentials and information you receive from your Relationship Manager.

Upload your files to Cross River

Refer to **prepare your input XML file** for how to prepare your files.

View files sent to you by Cross River

Refer to **output files** for an explanation of the files sent to you.

Reach out to **support** for any questions.

File processing

We will check for a new input file every 30 minutes. Acknowledgment files containing initial processing information are normally available within minutes.

4.6.3. Requirements per rail

This section describes the XML file requirements for the following payment methods (rails):

- **Book transfers**
- **ACH push CCD**
- **ACH push PPD**
- **Fedwire**
- **ACH direct debit**
- **RTP/FedNow**

For more information on the code structure for each payment method, see the corresponding topic in the XML input file examples section.

See **Prepare input XML file** for more details on the blocks and tags used in the XML examples below.

Book transfers

For book transfers (among 2 accounts within Cross River):

Both the **DbtrAgt** block and the **CdtrAgt** block must contain Cross River's routing number (021214891) in the **MmbId** element.

ACH push CCD

For ACH Push CCD transfers, the **PmtTpInf** block must be according to the following example:

```xml
<PmtTpInf>
        <InstrPrty>NORM</InstrPrty>
        <SvcLvl>
                <Cd>NURG</Cd>
        </SvcLvl>
        <LclInstrm>
                <Prtry>CCD</Prtry>
        </LclInstrm>
</PmtTpInf>
```

All the above fields are required. We validate for these fields in order to know what type of transaction you are requesting. If it can't find any of these, it will cause an error.

ACH push PPD

For ACH Push PPD transfers, the **PmtTpInf** block must be according to the following example:

```xml
<PmtTpInf>
        <InstrPrty>NORM</InstrPrty>
        <SvcLvl>
                <Cd>NURG</Cd>
        </SvcLvl>
        <LclInstrm>
                <Prtry>PPD</Prtry>
        </LclInstrm>
</PmtTpInf>
```

All the above fields are required. We validate for these fields in order to know what type of transaction you are requesting. If it can't find any of these, it will cause an error.

Fedwire

For Fedwire payments:

- The **CdtrAgt**'s **MmbId** value must be different than CR's routing number (021214891).

XML

```xml
<CdtrAgt>
          <FinInstnId>
              <ClrSysMmbId>
                  <ClrSysId>
                      <Cd>USABA</Cd>
                  </ClrSysId>
                  <MmbId>123456789</MmbId>
              </ClrSysMmbId>
              <Nm>THE BANK OF NEW YORK MELLON</Nm>
              <PstlAdr>
                 <Ctry>US</Ctry>
              </PstlAdr>
          </FinInstnId>
</CdtrAgt>
```

- The **PmtTpInf**'s **Cd** value must be "URGP".

XML

```xml
<PmtTpInf>
         <InstrPrty>NORM</InstrPrty>
         <SvcLvl>
             <Cd>URGP</Cd>
         </SvcLvl>
</PmtTpInf>
```

ACH direct debits (ACH Pull)

For ACH direct debits:

- The file type must be: XML pain.008.001.02 ISO 20022.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.008.001.02">
```

- The Payment Method (**PmtMtd**) value must be “DD”.

```xml
<PmtInf>
    ...
    <PmtMtd>DD</PmtMtd>
    ...
<PmtInf>
```

- The Creditor Agent (**CdtrAgt**)'s Member ID (**MmbId**) value must always be CR's routing number (021214891).

```xml
<CdtrAgt>
        <FinInstnId>
            <ClrSysMmbId>
                <MmbId>021214891</MmbId>
            </ClrSysMmbId>
            ...
        </FinInstnId>
</CdtrAgt>
```

- The Payment Type Information must include the Service Level Code of “NURG” and the Local Instrument Proprietary value, which is the ACH Pull type (“CCD” or “PPD”).

```xml
<PmtTpInf>
    <SvcLvl>
        <Cd>NURG</Cd>
  </SvcLvl>
  <LclInstrm>
    <Prtry>CCD</Prtry>
  </LclInstrm>
</PmtTpInf>
```

For example:

```xml
<PmtInf>
    ...
    <PmtMtd>DD</PmtMtd>
    ...
    <PmtTpInf>
    <SvcLvl>
        <Cd>NURG</Cd>
    </SvcLvl>
    <LclInstrm>
        <Prtry>CCD</Prtry>
    </LclInstrm>
    </PmtTpInf>
    ...
    <CdtrAgt>
    ...
    <MmbId>021214891</MmbId>
    ...
    </CdtrAgt>
    ...
</PmtInf>
```

All the fields shown above for Direct Debit are required so that we can identify the payment type and details.

RTP send (push) funds

This is how you should structure your XML for sending funds using RTP.

XML

```xml
<PmtTpInf>

<InstrPrty>NORM</InstrPrty>

<SvcLvl>

<Cd>URNS</Cd>

</SvcLvl>

<LclInstrm>

<Prtry>rtp</Prtry>

</LclInstrm>

</PmtTpInf>
```

4.6.4. Prepare input XML file

The XML Input files you send to Cross River for processing must include certain building blocks. These building blocks are part of the ISO 20022 standard. The following topics describe the generic building blocks and elements that CR needs you to include so we can process your files.

The first page under this topic describes the input **PAIN.001.001.03 input file**, which is used for Book (internal) transfers, ACH Push transfers, and Fedwire transfers.

The second page describes the input **PAIN.008.001.02 input file**, which is used for Direct Debit transfers.

> You can include up to 250,000 transactions in each file you send.

4.6.4.1. PAIN.001.001.03 input file

This page contains a description of the input XML file for the PAIN.001.001.03 format, which is used for Book transfers, ACH Push CCD/PPD, and Fedwires.

The following is a high-level example of a PAIN.001.001.03 input XML file:

XML

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <CstmrCdtTrfInitn>
        <GrpHdr>

          ...

        </GrpHdr>
        <PmtInf>
          ...
        </PmtInf>
        <PmtInf>
          ...
        </PmtInf>
    </CstmrCdtTrfInitn>
</Document>
```

The root document tag for the input file is CstmrCdtTrfInitn. It contains a Group Header and at least one Payment Information building block (corresponding to a batch). The Group Header contains metadata that relates to all the batches in the file. Each batch contains meta data for all the transactions within.

Group header (GrpHdr)

Your XML Input file must include a group header using the GrpHdr building block. This building block is only present once in a file and contains a set of characteristics shared by all the individual instructions included in the message.

```xml
<GrpHdr>
    <MsgId>ABCDEFG090301</MsgId>
    <CreDtTm>2013-08-28T17:12:44</CreDtTm>
    <NbOfTxs>5</NbOfTxs>
    <CtrlSum>43236.93</CtrlSum>
    <InitgPty>
      ...
    </InitgPty>
</GrpHdr>
```

Here's an explanation of the different tags in the GrpHdr block as shown above:

- **MsgId**: An ID number for the message, which you assign before sending the file to CRB. Best practice: Use a new for each file.**
- **CreDtTm**: The date and time the payment instruction was created.
- **NbOfTxs**: The total number of transaction instruction blocks in the message. Each instruction corresponds to one transaction, and will form a separate instruction block. For more information on the instruction blocks, see the Credit Transfer Transaction Information section below.
- **CtrlSum**: The total amount (as a number) of all the instructions included in the file, irrespective of currencies, used as a control sum. **Note**: Unicorn supports USD only.
- **InitgPty**: Indicates the party initiating the transfer (debtor). See the Initiating Party section that follows for more details on the tags inside the **InitgPty** block.

> All the above elements in the **GrpHdr** block are required.

Initiating party (InitgPty)

The Initiating Party is the party initiating the payment (the debtor).

```xml
<InitgPty>
                <Nm>John Doe Corporation</Nm>
                <Id>
                    <OrgId>
                        <BICOrBEI>JOHNDOE99</BICOrBEI>
                    </OrgId>
                </Id>
   </InitgPty>
```

Here's an explanation of the different tags in the InitgPty block as shown above:

- **Nm**: Name by which the initiating party is known and which is usually used to identify that party.
- **Id**: Identification. The parent element of the OrgId element containing the identifying information about the initiating party.
- **OrgId**: Organization Identification, containing the BICOrBEI code.
- **BICOrBEI**: Business Identification Code. A unique and unambiguous identifier of the initiating party, which can be an organization or an individual person.

> All the above elements in the **InitgPty** block are required.

Payment information (PmtInf)

The **PmtInf** block contains payment information per batch in your file. You must include at least one **PmtInf** block in your file. In most cases the input file contains only one **PmtInf** block, with one set of payment instructions. This enables you to indicate general properties (such as execution date, debtor, payment type and debited account) once at the level of the **PmtInf** block.

You might want to use multiple **PmtInf** blocks if the file includes instructions to debit more than one account. In that case, you need a **PmtInf** block for each account that is going to be debited.

```xml
<PmtInf>
    <PmtInfId>CASH10002273</PmtInfId>
    <PmtMtd>TRF</PmtMtd>
    <BtchBookg>false</BtchBookg>
    <NbOfTxs>1</NbOfTxs>
    <CtrlSum>0.01</CtrlSum>
    <PmtTpInf>
        ...
    </PmtTpInf>
    <ReqdExctnDt>2019-11-19</ReqdExctnDt>
    <Dbtr>
        ...
    </Dbtr>
    <DbtrAcct>
        ...
    </DbtrAcct>
    <DbtrAgt>
        ...
    </DbtrAgt>
    <ChrgBr>SHAR</ChrgBr>
    <CdtTrfTxInf>
        ...
    </CdtTrfTxInf>
</PmtInf>
```

Below is an explanation of the different top-level tags and blocks found in the **PmtInf** block as shown above. These tags and blocks appear once for each **PmtInf** block, except for the **CdtTrfTxInf** block, which can appear multiple times, representing multiple transactions. Each of the other tags and blocks applies to all **CdtTrfTxInf** blocks that appear in the **PmtInf** block:

- **PmtInf**: This block contains payment information, such as debtor and payment type information. You can use this block repeatedly within the same input file. **Note:** One or more instances of the PmtInf element is required.
- **PmtInfId**: The unique ID number for this batch, which is assigned by the originating party. **Note:** This element is required.
- **PmtMtd**: Payment Method. For credit transfers you should define it as "TRF". **Note:** This element is required.

- **BtchBookg**: Defines how CRB should handle the debit. If the tag is set to "TRUE", then all debit instructions will be handled as one consolidated debit. If the tag is set to "FALSE", it means that you want each debit to be handled separately. **Note**: Currently the system will always behave as if the value is "FALSE".
- **NbOfTxs**: The number of transactions within this batch. **Note:** This element is required.
- **CtrlSum**: The sum total of all instructions within this batch, irrespective of currencies, used as a control sum. **Note**: Unicorn currently supports "USD" only. **Note:** This element is required.
- **PmtTpInf**: The Payment Type Information block, including a priority level. See the Payment Type Information section below for more details on the **PmtTpInf** block. **Note**: This element is required.
- **ReqdExctnD**t: Requested Execution Date. The date on which the originator's account is to be debited. This tag currently supports current dates. Support for future dates will come in a future release. **Note**: This element is required.
- **Dbtr**: Debtor block. Contains the name and postal address of the originator. See the Debtor section below for an example of the Dbtr block. **Note**: This element is required.
- **DbtrAcct**: Debtor Account. The account of the originator that will be debited. See the Debtor Account section below for more details on the DbtrAcct block. **Note**: This element is required.
- **DbtrAgt**: Debtor Agent block. Details on the debtor's financial institution. See the Debtor Agent section below for more details on the DbtrAgt block. **Note**: This element is required.
- **ChrgBr**: ChargeBearer. Specifies the bearer of any transaction charges. Possible values: ‘CRED’ - All transaction charges are to be borne by the creditor. ‘DEBT’ - All transaction charges are to be borne by the debtor. ‘SHAR’ - Both parties bear their own charges. **Note**: This element is required. You should set the value to "SHAR".
- **CdtTrfTxInf**: Credit Transfer Transaction Information. Includes elements related to the credit side of the transaction, such as creditor and remittance information. This block can appear multiple times within the same PmtInf block. See the Credit Transfer Transaction Information section below for more details on the **CdtTrfTxInf** block. **Note**: One or more of the **CdtTrfTxInf** element is required.

Payment type information (PmtTpInf)

The PmtTpInf block contains information on the payment type. The following example is generic, while each example after it is specific to a rail type.

XML

```xml
<PmtTpInf>
    <InstrPrty>NORM</InstrPrty>
    <SvcLvl>
        <Cd>NURG</Cd>
    </SvcLvl>
</PmtTpInf>
```

For ACH Push CCD, the following **PmtTpInf** block is used:

XML

```xml
<PmtTpInf>
  <InstrPrty>NORM</InstrPrty>
  <SvcLvl>
    <Cd>NURG</Cd>
  </SvcLvl>
  <LclInstrm>
    <Prtry>CCD</Prtry>
  </LclInstrm>
</PmtTpInf>
```

For ACH Push PPD, the following **PmtTpInf** block is used:

XML

```xml
<PmtTpInf>
  <InstrPrty>NORM</InstrPrty>
  <SvcLvl>
    <Cd>NURG</Cd>
  </SvcLvl>
  <LclInstrm>
    <Prtry>PPD</Prtry>
  </LclInstrm>
</PmtTpInf>
```

For Fedwires and Book transfers, the following **PmtTpInf** block is used:

```xml
<PmtTpInf>
   <InstrPrty>NORM</InstrPrty>
   <SvcLvl>
     <Cd>URGP</Cd>
   </SvcLvl>
 </PmtTpInf>
```

Here's an explanation of the different tags in the PmtTpInf block as shown in the examples above.

- **InstrPrty**: Instruction Priority. Priority of the instruction (transaction request). Value reflects whether the payment processor is providing priority processing or normal processing. Currently "NORM" is the only available value. Note: This element is required.
- **SvcLvl**: Service Level. Contains the payment urgency level (**Cd**) in a child element. Note: This element is required.
- **Cd**: Code. The payment urgency Level. This element is required. **Note**: For Fedwires and Book transfers, the Cd value must be "URGP". For ACH Push CCD and PPD, the Cd value must be "NURG".
- **LclInstrm**: Local Instrument. Used to specify a local instrument, local clearing option and/or to further qualify the service or service level. This element is required for ACH Push CCD and PPD.
- **Prtry**: Proprietary. This element is required for ACH Push CCD and PPD. The **Prtry** value must be the ACH Push type ("CCD" or "PPD").

Debtor (Dbtr)

The **Dbtr** block contains information on the name and postal address of the originator (debtor).

```xml
<Dbtr>
    <Nm>John Doe Corporation</Nm>
    <PstlAdr>
        <StrtNm>999 Any Street</StrtNm>
        <PstCd>99999</PstCd>
        <TwnNm>Anytown</TwnNm>
        <CtrySubDvsn>CA</CtrySubDvsn>
        <Ctry>US</Ctry>
    </PstlAdr>
</Dbtr>
```

- **Nm**: Debtor name.
- **PstlAdr**: A block containing the postal address tags of the debtor.

All the above fields in the **Dbtr** block are required.

Debtor account (DbtrAcct)

The **DbtrAccount** block contains information on the account of the originator that will be debited.

```xml
<DbtrAcct>
    <Id>
        <Othr>
            <Id>2000526789</Id>
        </Othr>
    </Id>
    <Ccy>USD</Ccy>
</DbtrAcct>
```

Here's an explanation of the different tags in the **DbtrAccount** block as shown above:

- **Id**: The sub-block containing the debtor's account identification information.
- **Othr**: The sub-block containing the debtor's Id tag.

- **Id**: The unique identifier of the debtor's account.
- **Ccy**: Currency. The ISO currency code of the debtor's account. **Note**: This element is required.

All the above fields in the **DbtrAcct** block are required.

Debtor agent (DbtrAgt)

The **DbtrAgt** block contains information on the originator's financial institution.

XML

```xml
<DbtrAgt>
    <FinInstnId>
        <ClrSysMmbId>
            <ClrSysId>
                <Cd>USABA</Cd>
            </ClrSysId>
            <MmbId>123456789</MmbId>
        </ClrSysMmbId>
        <PstlAdr>
            <Ctry>US</Ctry>
        </PstlAdr>
    </FinInstnId>
</DbtrAgt>
```

Here's an explanation of the different tags in the **DbtrAgt** block as shown above:

- **FinInstnId**: Financial Institution Identification sub-block.
- **ClrSysMmbId**: Clearing System Member Identification sub-block. Contains information used to identify a member within a clearing system.
- **ClrSysId**: Clearing System Identification. Specification of the pre-agreed offering between clearing agents or the channel through which the payment instruction is processed.
- **Cd**: Code: Identifier of the debtor's clearing system, in a coded form as published in an external list.

- **MmbId**: Member Identification: Identification (routing number) of the debtor's financial institution in the debtor's clearing system. Note: For Book transfers, the value of this element must always be CR's routing number (021214891).
- **PstlAdr**: Postal Address sub-block.
- **Ctry**: Country code. The country code for the debtor's financial institution.

> All the above elements in the **DbtrAgt** block are required.

Credit transfer transaction information (CdtTrfTxInf)

The **CdtTrfTxInf** block includes elements related to the credit side of the transaction, such as creditor and remittance information for the transaction. You can use this block repeatedly within the same **PmtInf** block. The number of occurrences of the **CdtTrfTxInf** block within a file is indicated by the **NbOfTxs** field in the Group Header (**GrpHdr)**.

```xml
<CdtTrfTxInf>
    <PmtId>
        <InstrId>100ACH000019</InstrId>
        <EndToEndId>0000000000000003</EndToEndId>
    </PmtId>
    <Amt>
        <InstdAmt Ccy="USD">0.01</InstdAmt>
    </Amt>
    <CdtrAgt>
        ...
    </CdtrAgt>
    <Cdtr>
        <Nm>John Doe Funding LLC</Nm>
        <PstlAdr>
            <StrtNm>999 Any Avenue</StrtNm>
            <PstCd>10000</PstCd>
            <TwnNm>New York</TwnNm>
            <CtrySubDvsn>NY</CtrySubDvsn>
            <Ctry>US</Ctry>
        </PstlAdr>
    </Cdtr>
    <CdtrAcct>
        <Id>
            <Othr>
                <Id>0123456789</Id>
            </Othr>
        </Id>
        <Tp>
            <Cd>CACC</Cd>
        </Tp>
        <Ccy>USD</Ccy>
    </CdtrAcct>
  <RmtInf>
        <Ustrd>Testing</Ustrd>
  </RmtInf>
</CdtTrfTxInf>
```

Here's an explanation of the different tags in the **CdtTrfTxInf** block as shown above:

- **PmtId**: Payment Identification sub-block. Provides identifying information regarding the transaction in child elements. **Note**: This element is required.

- **InstrId**: Instruction Identification. Reference of the remitter, referring to a set of payments of which this payment is a part of. This information is not sent to the beneficiary.
- **EndToEndId**: End to End Identification: End-to-end reference number of the credit transfer. This information is sent to the beneficiary. **Note**: This element is required.
- **Amt**: Amount sub-block. Contains the transaction amount in a child element. **Note**: This element is required.
- **InstdAmt**: Instructed Amount. The amount of the credit transfer in the indicated currency. **Note**: This element is required.
- **CdtrAgt**: Creditor Agent sub-block. Contains details on the creditor's financial institution for the transaction. See the Creditor Agent section below for more details on the **CdtrAgt** block. **Note**: This element is required.
- **Cdtr**: The creditor sub-block. Contains details on the creditor for the transaction, including **Nm** (name) and **PstlAdr** (postal address) elements. **Note**: The Cdtr block and its **Nm** element are required. The **PstlAdr** element is not required.
- **CdtrAcct**: Creditor account sub-block for the transaction, containing the creditor account number and currency in its child elements. **Note**: This element is required.
- **Id:** Identification sub-block. Contains an identification of the creditor's account in child elements. **Note**: This element is required.
- **Othr**: Sub-block containing the creditor's Id tag. **Note**: This element is required.
- **Id**: The unique identifier of the creditor's account. **Note**: This element is required.
- **Tp**: Type. Contains the code in its child elements.
- **Cd**: Code. The code of the account type.
- **Ccy**: ISO currency code of the beneficiary account.
- **RmtInf**: The remittance information to send along with the transaction. **Note**: This element is required.
- **Ustrd**: Unstructured description of the transaction. Maximum limit of 50 characters including spaces.**Note**: This element is required.

Creditor agent (CdtrAgt)

The **CdtrAgt** block contains information on the creditor's financial institution for the transaction.

```xml
<CdtrAgt>
    <FinInstnId>
        <ClrSysMmbId>
            <ClrSysId>
                <Cd>USABA</Cd>
            </ClrSysId>
            <MmbId>123456789</MmbId>
        </ClrSysMmbId>
        <Nm>Cross River Bank NJ</Nm>
        <PstlAdr>
            <Ctry>US</Ctry>
        </PstlAdr>
    </FinInstnId>
</CdtrAgt>
```

Here's an explanation of the different tags in the **CdtrAgt** block as shown above:

- **FinInstnId**: Financial Institution Identification sub-block.
- **ClrSysMmbId**: Clearing System Member Identification sub-block. Contains information used to identify a member within a clearing system.
- **ClrSysId**: Clearing System Identification. Specification of the pre-agreed offering between clearing agents or the channel through which the payment instruction is processed.
- **Cd**: Code: Identification of the creditor's clearing system, in a coded form as published in an external list.
- **Nm**: Name of the creditor's financial institution.
- **MmbId**: Member Identification: Identification (routing number) of the creditor's financial institution in the creditor's clearing system. **Note**: For Book transfers, the value of this element must always be CR's routing number (021214891).
- **PstlAdr**: Postal Address sub-block.
- **Ctry**: Country code. The ISO country code for the creditor's financial institution.

> All the above elements in the **CdtrAgt** block are required.

> For a Fedwire transaction request, the **CdtrAgt** must not be Cross River.

All payment types example

Wrapping it all up, an example with all payment types

XML

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>DOMT11234562AllRails1</MsgId>
            <CreDtTm>2020-06-29T10:24:09</CreDtTm>
            <NbOfTxs>3</NbOfTxs>
            <CtrlSum>3.03</CtrlSum>
            <InitgPty>
                <Nm>Shach Corp</Nm>
                <Id>
                    <OrgId>
                        <BICOrBEI>THESHACH99</BICOrBEI>
                    </OrgId>
                </Id>
            </InitgPty>
        </GrpHdr>
        <PmtInf>
            <PmtInfId>DOMT12345678ACH1</PmtInfId>
            <PmtMtd>TRF</PmtMtd>
            <BtchBookg>false</BtchBookg>
            <NbOfTxs>1</NbOfTxs>
            <CtrlSum>0.01</CtrlSum>
            <PmtTpInf>
                <InstrPrty>NORM</InstrPrty>
                <SvcLvl>
                    <Cd>NURG</Cd>
                </SvcLvl>
                <LclInstrm>
                    <Prtry>CCD</Prtry>
                </LclInstrm>
            </PmtTpInf>
            <ReqdExctnDt>2020-06-29</ReqdExctnDt>
            <Dbtr>
                <Nm>John Doe Corporation</Nm>
                <PstlAdr>
                    <StrtNm>999 Battery Street, 13th Floor</StrtNm>
                    <PstCd>99999</PstCd>
                    <TwnNm>Anytown</TwnNm>
                    <CtrySubDvsn>CA</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Dbtr>
            <DbtrAcct>
                <Id>
```

```xml
                <Othr>
                    <Id>2756493439</Id>
                </Othr>
            </Id>
            <Ccy>USD</Ccy>
        </DbtrAcct>
        <DbtrAgt>
            <FinInstnId>
                <ClrSysMmbId>
                    <ClrSysId>
                        <Cd>USABA</Cd>
                    </ClrSysId>
                    <MmbId>123456789</MmbId>
                </ClrSysMmbId>
                <PstlAdr>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </FinInstnId>
        </DbtrAgt>
        <ChrgBr>SHAR</ChrgBr>
        <CdtTrfTxInf>
            <PmtId>
                <InstrId>234ACHC123455ACH1</InstrId>
                <EndToEndId>0000000000000011ACH1</EndToEndId>
            </PmtId>
            <Amt>
                <InstdAmt Ccy="USD">0.01</InstdAmt>
            </Amt>
            <CdtrAgt>
                <FinInstnId>
                    <ClrSysMmbId>
                        <ClrSysId>
                            <Cd>USABA</Cd>
                        </ClrSysId>
                        <MmbId>123456789</MmbId>
                    </ClrSysMmbId>
                    <Nm>Cross River Bank NJ</Nm>
                    <PstlAdr>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </FinInstnId>
            </CdtrAgt>
            <Cdtr>
                <Nm>John Doe Funding LLC</Nm>
                <PstlAdr>
                    <StrtNm>999 Any Avenue</StrtNm>
```

```
                    <PstCd>10000</PstCd>
                    <TwnNm>New York</TwnNm>
                    <CtrySubDvsn>NY</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Cdtr>
            <CdtrAcct>
                <Id>
                    <Othr>
                        <Id>0123456789</Id>
                    </Othr>
                </Id>
                <Tp>
                    <Cd>CACC</Cd>
                </Tp>
                <Ccy>USD</Ccy>
            </CdtrAcct>
            <RmtInf>
                <Ustrd>Testing</Ustrd>
            </RmtInf>
        </CdtTrfTxInf>
    </PmtInf>
    <PmtInf>
        <PmtInfId>DOMT10000000wire1</PmtInfId>
        <PmtMtd>TRF</PmtMtd>
        <BtchBookg>false</BtchBookg>
        <NbOfTxs>1</NbOfTxs>
        <CtrlSum>3.00</CtrlSum>
        <PmtTpInf>
            <InstrPrty>NORM</InstrPrty>
            <SvcLvl>
                <Cd>URGP</Cd>
            </SvcLvl>
        </PmtTpInf>
        <ReqdExctnDt>2020-07-15</ReqdExctnDt>
        <Dbtr>
            <Nm>John Doe Corporation</Nm>
            <PstlAdr>
                <StrtNm>999 Any Street, 13th Floor</StrtNm>
                <PstCd>99999</PstCd>
                <TwnNm>Anytown</TwnNm>
                <CtrySubDvsn>CA</CtrySubDvsn>
                <Ctry>US</Ctry>
            </PstlAdr>
        </Dbtr>
        <DbtrAcct>
```

```xml
                <Id>
                    <Othr>
                        <Id>2756493439</Id>
                    </Othr>
                </Id>
                <Ccy>USD</Ccy>
            </DbtrAcct>
            <DbtrAgt>
                <FinInstnId>
                    <ClrSysMmbId>
                        <ClrSysId>
                            <Cd>USABA</Cd>
                        </ClrSysId>
                        <MmbId>123456789</MmbId>
                    </ClrSysMmbId>
                    <PstlAdr>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </FinInstnId>
            </DbtrAgt>
            <ChrgBr>SHAR</ChrgBr>
            <CdtTrfTxInf>
                <PmtId>
                    <InstrId>100FED000000xxxwire1</InstrId>
                    <EndToEndId>0000000000000065xxxwire1</EndToEndId>
                </PmtId>
                <Amt>
                    <InstdAmt Ccy="USD">3.00</InstdAmt>
                </Amt>
                <CdtrAgt>
                    <FinInstnId>
                        <ClrSysMmbId>
                            <ClrSysId>
                                <Cd>USABA</Cd>
                            </ClrSysId>
                            <MmbId>123456789</MmbId>
                        </ClrSysMmbId>
                        <Nm>Cross River Bank NJ</Nm>
                        <PstlAdr>
                            <Ctry>US</Ctry>
                        </PstlAdr>
                    </FinInstnId>
                </CdtrAgt>
                <Cdtr>
                    <Nm>John Doe Funding LLC</Nm>
                    <PstlAdr>
```

```
                    <StrtNm>999 Any Avenue</StrtNm>
                    <PstCd>10000</PstCd>
                    <TwnNm>New York</TwnNm>
                    <CtrySubDvsn>NY</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Cdtr>
            <CdtrAcct>
                <Id>
                    <Othr>
                        <Id>0123456789</Id>
                    </Othr>
                </Id>
                <Ccy>USD</Ccy>
            </CdtrAcct>
            <RmtInf>
                <Ustrd>AllRails Wire 1</Ustrd>
            </RmtInf>
        </CdtTrfTxInf>
    </PmtInf>
    <PmtInf>
        <PmtInfId>DOMT12345678RTP1</PmtInfId>
        <PmtMtd>TRF</PmtMtd>
        <BtchBookg>false</BtchBookg>
        <NbOfTxs>1</NbOfTxs>
        <CtrlSum>0.02</CtrlSum>
        <PmtTpInf>
            <InstrPrty>NORM</InstrPrty>
            <SvcLvl>
                <Cd>URNS</Cd>
            </SvcLvl>
            <LclInstrm>
                <Prtry>RTP</Prtry>
            </LclInstrm>
        </PmtTpInf>
        <ReqdExctnDt>2020-06-29</ReqdExctnDt>
        <Dbtr>
            <Nm>John Doe Corporation</Nm>
            <PstlAdr>
                <StrtNm>999 Battery Street, 13th Floor</StrtNm>
                <PstCd>99999</PstCd>
                <TwnNm>Anytown</TwnNm>
                <CtrySubDvsn>CA</CtrySubDvsn>
                <Ctry>US</Ctry>
            </PstlAdr>
        </Dbtr>
```

```xml
<DbtrAcct>
    <Id>
        <Othr>
            <Id>2756493439</Id>
        </Othr>
    </Id>
    <Ccy>USD</Ccy>
</DbtrAcct>
<DbtrAgt>
    <FinInstnId>
        <ClrSysMmbId>
            <ClrSysId>
                <Cd>USABA</Cd>
            </ClrSysId>
            <MmbId>123456789</MmbId>
        </ClrSysMmbId>
        <PstlAdr>
            <Ctry>US</Ctry>
        </PstlAdr>
    </FinInstnId>
</DbtrAgt>
<ChrgBr>SHAR</ChrgBr>
<CdtTrfTxInf>
    <PmtId>
        <InstrId>234ACHC123455RTP1</InstrId>
        <EndToEndId>0000000000000011RTP1</EndToEndId>
    </PmtId>
    <Amt>
        <InstdAmt Ccy="USD">0.02</InstdAmt>
    </Amt>
    <CdtrAgt>
        <FinInstnId>
            <ClrSysMmbId>
                <ClrSysId>
                    <Cd>USABA</Cd>
                </ClrSysId>
                <MmbId>123456789</MmbId>
            </ClrSysMmbId>
            <Nm>Cross River Bank NJ</Nm>
            <PstlAdr>
                <Ctry>US</Ctry>
            </PstlAdr>
        </FinInstnId>
    </CdtrAgt>
    <Cdtr>
        <Nm>John Doe Funding LLC</Nm>
```

```
                <PstlAdr>
                    <StrtNm>999 Any Avenue</StrtNm>
                    <PstCd>10000</PstCd>
                    <TwnNm>New York</TwnNm>
                    <CtrySubDvsn>NY</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Cdtr>
            <CdtrAcct>
                <Id>
                    <Othr>
                        <Id>0123456789</Id>
                    </Othr>
                </Id>
                <Tp>
                    <Cd>CACC</Cd>
                </Tp>
                <Ccy>USD</Ccy>
            </CdtrAcct>
            <RmtInf>
                <Ustrd>AllRails RTP 1</Ustrd>
            </RmtInf>
        </CdtTrfTxInf>
    </RmtInf>
```

4.6.4.2. PAIN.008.001.02 input file

This page contains a description of the input XML file for the PAIN.008.001.02 specification, which is used for ACH Direct Debit (ACH Pull) CCD and PPD payments.

The following is a high-level example of a PAIN.008.001.02 input XML file:

XML

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.008.001.02">
    <CstmrDrctDbtInitn>
        <GrpHdr>
            ...
        </GrpHdr>
        <PmtInf>
           ...
        </PmtInf>
    </CstmrDrctDbtInitn>
</Document>
```

The root document tag for the input file is **CstmrDrctDbtInitn**. It contains a Group Header and at least one Payment Information building block (corresponding to a batch). The Group Header contains metadata that relates to all the batches in the file. Each batch contains meta data for all the transactions within.

Group header (GrpHdr)

Your XML Input file must include a group header using the GrpHdr building block. This building block is only present once in a file and contains a set of characteristics shared by all the individual instructions included in the message.

```xml
<GrpHdr>
    <MsgId>ABCDEFG090301</MsgId>
    <CreDtTm>2013-08-28T17:12:44</CreDtTm>
    <NbOfTxs>5</NbOfTxs>
    <CtrlSum>43236.93</CtrlSum>
    <InitgPty>
      ...
    </InitgPty>
</GrpHdr>
```

Here's an explanation of the different tags in the GrpHdr block as shown above:

- **MsgId**: An ID number for the message, which you assign before sending the file to CR. **Best practice**: Use a new for each file.**
- **CreDtTm**: The date and time the payment instruction was created.
- **NbOfTxs**: The total number of transaction instruction blocks in the message. Each instruction corresponds to one transaction, and will form a separate instruction block. For more information on the transaction instruction blocks, see the Direct Debit Transaction Information section below.
- **CtrlSum**: The total amount (as a number) of all the instructions included in the file, irrespective of currencies, used as a control sum. **Note**: Unicorn currently supports USD only.
- **InitgPty**: Indicates the party initiating the transfer. This is the party initiating the credit transfer on behalf of the debtor. See the Initiating Party section that follows for more details on the tags inside the **InitgPty** block.

> All the above elements in the **GrpHdr** block are required.

Initiating party (InitgPty)

The Initiating Party is the party initiating the payment. This is the party that initiates the credit transfer on behalf of the debtor.

```xml
<InitgPty>
    <Nm>John Doe Corporation</Nm>
    <Id>
        <OrgId>
            <Othr>
                <Id>0123456789</Id>
            </Othr>
        </OrgId>
    </Id>
</InitgPty>
```

Here's an explanation of the different tags in the **InitgPty** block as shown above:

Nm: Name by which the originating party is known and which is usually used to identify that party.

Id: Identifier. The parent element of the **OrgId** element containing the identifying information about the initiating party.

OrgId: Organization Identification block containing the initiating party's identification in its child elements.

Othr: A block containing the initiating party's identification in a child element.

Id: The unique and unambiguous identifier the initiating party, which can be an organization or an individual person.

> All the above elements in the **InitgPty** block are required.

Payment information (PmtInf)

The **PmtInf** block contains payment information per batch in your file. You must include at least one **PmtInf** block in your file. In most cases the input file contains only one **PmtInf** block, with one set of payment instructions. This enables you to indicate general properties

(such as execution date, creditor information, and credited account) once at the level of the **PmtInf** block.

You might want to use multiple **PmtInf** blocks if the file includes instructions to credit more than one account. In that case, you need a **PmtInf** block for each account that is going to be credited.

XML

```xml
<PmtInf>
    <PmtInfId>DOMC10000025</PmtInfId>
    <PmtMtd>DD</PmtMtd>
    <BtchBookg>false</BtchBookg>
    <NbOfTxs>5</NbOfTxs>
    <CtrlSum>12.01</CtrlSum>
    <PmtTpInf>
        ...
    </PmtTpInf>
    <ReqdColltnDt>2020-08-21</ReqdColltnDt>
    <Cdtr>
        ...
    </Cdtr>
    <CdtrAcct>
        ...
    </CdtrAcct>
    <CdtrAgt>
        ...
    </CdtrAgt>
    <DrctDbtTxInf>
        ...
    </DrctDbtTxInf>
</PmtInf>
```

Below is an explanation of the different top-level tags and blocks found in the **PmtInf** block as shown above. These tags and blocks appear once for each **PmtInf** block, except for the **DrctDbtTxInf** block, which can appear multiple times, representing multiple transactions. Each of the other tags and blocks applies to all **DrctDbtTxInf** blocks that appear in the **PmtInf** block:

- **PmtInf**: This block contains payment information, such as creditor and payment type information. You can use this block repeatedly within the same input file. **Note**: One

or more instances of the **PmtInf** element is required.

- **PmtInfId**: The unique ID number for this batch, which is assigned by the originating party. **Note**: This element is required.
- **PmtMtd**: Payment Method. For direct debit transactions you should define it as "DD". Note: This element is required.
- **BtchBookg**: Defines how CR should handle the debit. If the tag is set to "TRUE", then all debit instructions will be handled as one consolidated debit. If the tag is set to "FALSE", it means that you want each debit to be handled separately. **Note**: Currently the system will always behave as if the value is "FALSE".
- **NbOfTxs**: The number of transactions within this batch. **Note**: This element is required.
- **CtrlSum**: The sum total of all instructions within this batch, irrespective of currencies, used as a control sum. **Note**: Unicorn currently supports "USD" only. **Note**: This element is required.
- **PmtTpInf**: The Payment Type Information block, including a priority level. See the Payment Type Information section below for more details on the **PmtTpInf** block. **Note**: This element is required.
- **ReqdCollItnDt**: Requested Collection Date. The date on which the originator's account is to be debited. This tag currently supports current dates. Support for future dates will come in a future release. **Note**: This element is required.
- **Cdtr**: Creditor block. Contains the name and postal address of the originator. See the Creditor section below for an example of the **Cdtr** block. **Note**: This element is required.
- **CdtrAcct**: Creditor Account. The account of the originator that will be credited. See the Creditor Account section below for more details on the **CdtrAcct** block. **Note**: This element is required.
- **CdtrAgt**: Creditor Agent block. Details on the creditor's financial institution. See the Creditor Agent section below for more details on the **CdtrAgt** block. **Note**: This element is required.
- **DrctDbtTxInf:** Direct Debit Transaction Information. Includes elements related to the debit side of the transaction, such as debtor and remittance information. This block can appear multiple times within the same **PmtInf** block. See the Direct Debit Transaction Information section below for more details on the **DrctDbtTxInf** block. **Note**: One or more of the **DrctDbtTxInf** element is required.

Payment type information (PmtTpInf)

The **PmtTpInf** block contains information on the payment type.

```XML
<PmtTpInf>
    <SvcLvl>
        <Cd>NURG</Cd>
    </SvcLvl>
    <LclInstrm>
        <Prtry>CCD</Prtry>
    </LclInstrm>
</PmtTpInf>
```

Here's an explanation of the different tags in the **PmtTpInf** block as shown in the examples above:

- **SvcLvl**: Service Level. Contains the payment urgency level (**Cd**) in a child element.
- **Cd**: Code: Payment urgency level. **Note:** This element is required. For direct debit transactions this has a fixed value of "NURG".
- **LclInstrm**: Local Instrument. Used to specify a local instrument, local clearing option and/or to further qualify the service or service level. **Note**: This element is required.
- **Prtry**: Proprietary. Note: This element is required. The value must be the ACH Pull type ("CCD" or "PPD").

Creditor (Cdtr)

The **Cdtr** block contains information on the name, postal address and ID of the originator (creditor).

```xml
<Cdtr>
    <Nm>John Doe Corporation</Nm>
    <PstlAdr>
        <Ctry>US</Ctry>
        <AdrLine>999 Any Street, 13th Floor</AdrLine>
        <AdrLine>99999 Anytown</AdrLine>
    </PstlAdr>
    <Id>
        <OrgId>
            <Othr>
                <Id>0123456789</Id>
            </Othr>
        </OrgId>
    </Id>
</Cdtr>
```

- **Nm**: Creditor name.
- **PstlAdr**: A block containing the postal address of the creditor, including country and address lines.
- **Id**: Identification block, containing information used to identify the creditor in child elements.
- **OrgId**: Organization Identification block containing the creditor identification in its child elements.
- **Othr**: A block containing the creditor identification in a child element.
- **Id**: A unique and unambiguous identifier of the creditor. This ID is identical to the Id field in the **InitgPty** block described above.

Creditor account (CdtrAcct)

The **CdtrAccount** block contains information on the account of the originator that will be credited.

```xml
<CdtrAcct>
    <Id>
        <Othr>
            <Id>0123456789</Id>
        </Othr>
    </Id>
    <Ccy>USD</Ccy>
</CdtrAcct>
```

Here's an explanation of the different tags in the **CdtrAccount** block as shown above:

- **Id**: The sub-block containing the creditor's account identification information.
- **Othr**: The sub-block containing the creditor's Id tag.
- **Id**: The unique identifier of the creditor's account.
- **Ccy**: Currency. The ISO currency code of the debtor's account.

> All the above elements in the **CdtrAcct** block are required.

Creditor agent (CdtrAgt)

The **CdtrAgt** block contains information on the originator's financial institution.

```xml
<CdtrAgt>
    <FinInstnId>
        <ClrSysMmbId>
            <MmbId>123456789</MmbId>
        </ClrSysMmbId>
        <PstlAdr>
            <Ctry>US</Ctry>
        </PstlAdr>
    </FinInstnId>
</CdtrAgt>
```

Here's an explanation of the different tags in the **CdtrAgt** block as shown above:

- **FinInstnId**: Financial Institution Identification sub-block.
- **ClrSysMmbId**: Clearing System Member Identification sub-block. Contains Information used to identify a member within a clearing system.
- **MmbId**: Member Identification: Identifier (routing number) of the creditor's financial institution in the creditor's clearing system. For Direct Debit, this value must always be CR's routing number (021214891).
- **PstlAdr**: Postal Address sub-block.
- **Ctry**: Country code. The country code for the creditor's financial institution.

> All the above elements in the **CdtrAgt** block are required.

Direct debit transaction information (DrctDbtTxInf)

The **DrctDbtTxInf** block includes elements related to the debit side of the transaction, such as debitor and remittance information for the transaction. You can use this block repeatedly within the same PmtInf block. The number of occurrences of the **DrctDbtTxInf** block within a file is indicated by the **NbOfTxs** field in the Group Header (**GrpHdr**).

XML

```xml
<DrctDbtTxInf>
    <PmtId>
        <EndToEndId>100DDEB000000</EndToEndId>
    </PmtId>
    <InstdAmt Ccy="USD">0.01</InstdAmt>
    <DbtrAgt>
        <FinInstnId>
            <ClrSysMmbId>
                <MmbId>123456789</MmbId>
            </ClrSysMmbId>
            <Nm>DUMMY</Nm>
            <PstlAdr>
                <Ctry>US</Ctry>
            </PstlAdr>
        </FinInstnId>
    </DbtrAgt>
    <Dbtr>
        <Nm>John Doe</Nm>
        <PstlAdr>
            <Ctry>US</Ctry>
        </PstlAdr>
    </Dbtr>
    <DbtrAcct>
        <Id>
            <Othr>
                    <Id>01234567890</Id>
            </Othr>
        </Id>
    </DbtrAcct>
    <RmtInf>
        <Ustrd>Testing</Ustrd>
    </RmtInf>
</DrctDbtTxInf>
```

Here's an explanation of the different tags in the **DrctDbtTxInf** block as shown above:

- **PmtId**: Payment Identification sub-block. Provides identifying information regarding the transaction in child elements. **Note**: This element is required.
- **EndToEndId**: End to End Identification: End-to-end reference number of the credit transfer. This information is sent to the beneficiary. **Note**: This element is required.

- **InstdAmt**: Instructed Amount. The amount of the credit transfer in the indicated currency. **Note**: This element is required.
- **DbtrAgt**: Debtor Agent block. Details on the debtor's financial institution for the transaction. See the Debtor Agent section below for more details on the **DbtrAgt** block. **Note**: This element is required.
- **Dbtr**: The debtor sub-block. Contains details on the debtor for the transaction, including **Nm** (name) and **PstlAdr** (postal address) elements. **Note**: The **Dbtr** block and its **Nm** element are required. The **PstlAdr** element is not required.
- **DbtrAcct**: Debtor account sub-block for the transaction, containing the debtor account number in its child elements. **Note**: This element is required.
- **Id**: Identification sub-block. Contains an identification of the debtor account in child elements. **Note**: This element is required.
- **Othr**: Sub-block containing the debtor's Id tag. **Note**: This element is required.
- **Id**: The unique identifier of the debtor's account. **Note**: This element is required.
- **RmtInf**: The remittance information to send along with the transaction. **Note**: This element is required for ACH Direct Debit.
- **Ustrd**: Unstructured description of the transaction. **Note**: This element is required.

Debtor agent (DbtrAgt)

The **DbtrAgt** block contains information on the debtor's financial institution.

XML

```xml
<DbtrAgt>
    <FinInstnId>
        <ClrSysMmbId>
            <MmbId>123456789</MmbId>
        </ClrSysMmbId>
        <Nm>John Doe Bank</Nm>
        <PstlAdr>
            <Ctry>US</Ctry>
        </PstlAdr>
    </FinInstnId>
</DbtrAgt>
```

Here's an explanation of the different tags in the **DbtrAg**t block as shown above:

- **FinInstnId**: Financial Institution Identification sub-block.
- **ClrSysMmbId**: Clearing System Member Identification sub-block. Contains Information used to identify a member within a clearing system.
- **MmbId**: Member Identification: Identification (routing number) of the debtor's financial institution in the debtor's clearing system.
- **Nm**: Name of the debtor's financial institution.
- **PstlAdr**: Postal Address sub-block.
- **Ctry**: Country code. The ISO country code for the debtor's financial institution.

> All the above elements in the **DbtrAgt** block are required.

4.6.5. XML input file examples

Cross River have these XML input file examples:

- **Book transfer**
- **ACH CCD/PPD**
- **Fedwire**
- **ACH direct debit**
- **RTP/FedNow**

4.6.5.1. Book transfer

> For Book Transfers the PAIN standard is **PAIN.001.001.03**.

XML

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>CASH12345678</MsgId>
            <CreDtTm>2020-07-29T18:32:51</CreDtTm>
            <NbOfTxs>2</NbOfTxs>
            <CtrlSum>2.04</CtrlSum>
            <InitgPty>
                <Nm>John Doe Corporation</Nm>
                <Id>
                    <OrgId>
                        <BICOrBEI>JOHNDOE99</BICOrBEI>
                    </OrgId>
                </Id>
            </InitgPty>
        </GrpHdr>
        <PmtInf>
            <PmtInfId>CASH12345678</PmtInfId>
            <PmtMtd>TRF</PmtMtd>
            <BtchBookg>true</BtchBookg>
            <NbOfTxs>2</NbOfTxs>
            <CtrlSum>2.04</CtrlSum>
            <PmtTpInf>
                <InstrPrty>NORM</InstrPrty>
                <SvcLvl>
                    <Cd>URGP</Cd>
                </SvcLvl>
            </PmtTpInf>
            <ReqdExctnDt>2020-07-29</ReqdExctnDt>
            <Dbtr>
                <Nm>John Doe Corporation</Nm>
                <PstlAdr>
                    <StrtNm>999 Any Street, 13th Floor</StrtNm>
                    <PstCd>99999</PstCd>
                    <TwnNm>Anytown</TwnNm>
                    <CtrySubDvsn>CA</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Dbtr>
            <DbtrAcct>
                <Id>
                    <Othr>
                        <Id>0123456789</Id>
                    </Othr>
```

```xml
                </Id>
                <Ccy>USD</Ccy>
            </DbtrAcct>
            <DbtrAgt>
                <FinInstnId>
                    <ClrSysMmbId>
                        <ClrSysId>
                            <Cd>USABA</Cd>
                        </ClrSysId>
                        <MmbId>123456789</MmbId>
                    </ClrSysMmbId>
                    <PstlAdr>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </FinInstnId>
            </DbtrAgt>
            <ChrgBr>SHAR</ChrgBr>
            <CdtTrfTxInf>
                <PmtId>
                    <InstrId>100BKTW004345</InstrId>
                    <EndToEndId>0000000000000072</EndToEndId>
                </PmtId>
                <Amt>
                    <InstdAmt Ccy="USD">0.04</InstdAmt>
                </Amt>
                <CdtrAgt>
                    <FinInstnId>
                        <ClrSysMmbId>
                            <ClrSysId>
                                <Cd>USABA</Cd>
                            </ClrSysId>
                            <MmbId>123456789</MmbId>
                        </ClrSysMmbId>
                        <Nm>Cross River Bank NJ</Nm>
                        <PstlAdr>
                            <Ctry>US</Ctry>
                        </PstlAdr>
                    </FinInstnId>
                </CdtrAgt>
                <Cdtr>
                    <Nm>John Doe Corporation</Nm>
                    <PstlAdr>
                        <StrtNm>999 Any Street, 13th Floor</StrtNm>
                        <PstCd>99999</PstCd>
                        <TwnNm>Anytown</TwnNm>
                        <CtrySubDvsn>CA</CtrySubDvsn>
```

```
                <Ctry>US</Ctry>
            </PstlAdr>
            <Id>
                <OrgId>
                    <Othr>
                        <Id>0123456789</Id>
                    </Othr>
                </OrgId>
            </Id>
        </Cdtr>
        <CdtrAcct>
            <Id>
                <Othr>
                    <Id>0123456789</Id>
                </Othr>
            </Id>
            <Ccy>USD</Ccy>
        </CdtrAcct>
        <RmtInf>
            <Ustrd>Testing</Ustrd>
        </RmtInf>
    </CdtTrfTxInf>
    <CdtTrfTxInf>
        <PmtId>
            <InstrId>100BKTW0000</InstrId>
            <EndToEndId>0000000000000073</EndToEndId>
        </PmtId>
        <Amt>
            <InstdAmt Ccy="USD">2.00</InstdAmt>
        </Amt>
        <CdtrAgt>
            <FinInstnId>
                <ClrSysMmbId>
                    <ClrSysId>
                        <Cd>USABA</Cd>
                    </ClrSysId>
                    <MmbId>123456789</MmbId>
                </ClrSysMmbId>
                <Nm>Cross River Bank NJ</Nm>
                <PstlAdr>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </FinInstnId>
        </CdtrAgt>
        <Cdtr>
            <Nm>John Doe Corporation</Nm>
```

```xml
                    <PstlAdr>
                        <StrtNm>999 Any Street, 13th Floor</StrtNm>
                        <PstCd>99999</PstCd>
                        <TwnNm>Anytown</TwnNm>
                        <CtrySubDvsn>CA</CtrySubDvsn>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                    <Id>
                        <OrgId>
                            <Othr>
                                <Id>0123456789</Id>
                            </Othr>
                        </OrgId>
                    </Id>
                </Cdtr>
                <CdtrAcct>
                    <Id>
                        <Othr>
                            <Id>0123456789</Id>
                        </Othr>
                    </Id>
                    <Ccy>USD</Ccy>
                </CdtrAcct>
                <RmtInf>
                    <Ustrd>Testing</Ustrd>
                </RmtInf>
            </CdtTrfTxInf>
        </PmtInf>
    </CstmrCdtTrfInitn>
```

4.6.5.2. ACH CCD/PPD

For ACH/CCD/PPD the PAIN standard is **PAIN.001.001.03**.

The example below is for the ACH Push CCD rail. The XML for ACH Push PPD is identical to that of ACH Push CCD, except that the value of the **Prtry** element is "PPD".

XML

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>DOMT11234562</MsgId>
            <CreDtTm>2020-06-29T10:24:09</CreDtTm>
            <NbOfTxs>1</NbOfTxs>
            <CtrlSum>0.01</CtrlSum>
            <InitgPty>
                <Nm>John Doe Corporation</Nm>
                <Id>
                    <OrgId>
                        <BICOrBEI>JOHNDOE99</BICOrBEI>
                    </OrgId>
                </Id>
            </InitgPty>
        </GrpHdr>
        <PmtInf>
            <PmtInfId>DOMT12345678</PmtInfId>
            <PmtMtd>TRF</PmtMtd>
            <BtchBookg>false</BtchBookg>
            <NbOfTxs>1</NbOfTxs>
            <CtrlSum>0.01</CtrlSum>
            <PmtTpInf>
                <InstrPrty>NORM</InstrPrty>
                <SvcLvl>
                    <Cd>NURG</Cd>
                </SvcLvl>
                <LclInstrm>
                    <Prtry>CCD</Prtry>
                </LclInstrm>
            </PmtTpInf>
            <ReqdExctnDt>2020-06-29</ReqdExctnDt>
            <Dbtr>
                <Nm>John Doe Corporation</Nm>
                <PstlAdr>
                    <StrtNm>999 Battery Street, 13th Floor</StrtNm>
                    <PstCd>99999</PstCd>
                    <TwnNm>Anytown</TwnNm>
                    <CtrySubDvsn>CA</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Dbtr>
            <DbtrAcct>
                <Id>
```

```xml
                <Othr>
                    <Id>0123456789</Id>
                </Othr>
            </Id>
            <Ccy>USD</Ccy>
        </DbtrAcct>
        <DbtrAgt>
            <FinInstnId>
                <ClrSysMmbId>
                    <ClrSysId>
                        <Cd>USABA</Cd>
                    </ClrSysId>
                    <MmbId>123456789</MmbId>
                </ClrSysMmbId>
                <PstlAdr>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </FinInstnId>
        </DbtrAgt>
        <ChrgBr>SHAR</ChrgBr>
        <CdtTrfTxInf>
            <PmtId>
                <InstrId>234ACHC123455</InstrId>
                <EndToEndId>0000000000000011</EndToEndId>
            </PmtId>
            <Amt>
                <InstdAmt Ccy="USD">0.01</InstdAmt>
            </Amt>
            <CdtrAgt>
                <FinInstnId>
                    <ClrSysMmbId>
                        <ClrSysId>
                            <Cd>USABA</Cd>
                        </ClrSysId>
                        <MmbId>123456789</MmbId>
                    </ClrSysMmbId>
                    <Nm>Cross River Bank NJ</Nm>
                    <PstlAdr>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </FinInstnId>
            </CdtrAgt>
            <Cdtr>
                <Nm>John Doe Funding LLC</Nm>
                <PstlAdr>
                    <StrtNm>999 Any Avenue</StrtNm>
```

```xml
                        <PstCd>10000</PstCd>
                        <TwnNm>New York</TwnNm>
                        <CtrySubDvsn>NY</CtrySubDvsn>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </Cdtr>
                <CdtrAcct>
                    <Id>
                        <Othr>
                            <Id>0123456789</Id>
                        </Othr>
                    </Id>
                    <Tp>
                        <Cd>CACC</Cd>
                    </Tp>
                    <Ccy>USD</Ccy>
                </CdtrAcct>
                <RmtInf>
                    <Ustrd>Testing</Ustrd>
                </RmtInf>
            </CdtTrfTxInf>
        </PmtInf>
    </CstmrCdtTrfInitn>
</Document>
```

4.6.5.3. Fedwire

> For Fedwire the PAIN standard is **PAIN.001.001.03**.

XML

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>DOMT10001174</MsgId>
            <CreDtTm>2020-07-15T16:20:05</CreDtTm>
            <NbOfTxs>3</NbOfTxs>
            <CtrlSum>6.01</CtrlSum>
            <InitgPty>
                <Nm>John Doe Corporation</Nm>
                <Id>
                    <OrgId>
                        <BICOrBEI>JOHNDOE99</BICOrBEI>
                    </OrgId>
                </Id>
            </InitgPty>
        </GrpHdr>
        <PmtInf>
            <PmtInfId>DOMT10000000</PmtInfId>
            <PmtMtd>TRF</PmtMtd>
            <BtchBookg>false</BtchBookg>
            <NbOfTxs>3</NbOfTxs>
            <CtrlSum>6.01</CtrlSum>
            <PmtTpInf>
                <InstrPrty>NORM</InstrPrty>
                <SvcLvl>
                    <Cd>URGP</Cd>
                </SvcLvl>
            </PmtTpInf>
            <ReqdExctnDt>2020-07-15</ReqdExctnDt>
            <Dbtr>
                <Nm>John Doe Corporation</Nm>
                <PstlAdr>
                    <StrtNm>999 Any Street, 13th Floor</StrtNm>
                    <PstCd>99999</PstCd>
                    <TwnNm>Anytown</TwnNm>
                    <CtrySubDvsn>CA</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Dbtr>
            <DbtrAcct>
                <Id>
                    <Othr>
                        <Id>0123456789</Id>
                    </Othr>
```

```xml
                </Id>
                <Ccy>USD</Ccy>
            </DbtrAcct>
            <DbtrAgt>
                <FinInstnId>
                    <ClrSysMmbId>
                        <ClrSysId>
                            <Cd>USABA</Cd>
                        </ClrSysId>
                        <MmbId>123456789</MmbId>
                    </ClrSysMmbId>
                    <PstlAdr>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </FinInstnId>
            </DbtrAgt>
            <ChrgBr>SHAR</ChrgBr>
            <CdtTrfTxInf>
                <PmtId>
                    <InstrId>100FED011119</InstrId>
                    <EndToEndId>0000000000000064</EndToEndId>
                </PmtId>
                <Amt>
                    <InstdAmt Ccy="USD">1.00</InstdAmt>
                </Amt>
                <CdtrAgt>
                    <FinInstnId>
                        <ClrSysMmbId>
                            <ClrSysId>
                                <Cd>USABA</Cd>
                            </ClrSysId>
                            <MmbId>123456789</MmbId>
                        </ClrSysMmbId>
                        <Nm>THE BANK OF NEW YORK MELLON</Nm>
                        <PstlAdr>
                            <Ctry>US</Ctry>
                        </PstlAdr>
                    </FinInstnId>
                </CdtrAgt>
                <Cdtr>
                    <Nm>John Doe Funding LLC</Nm>
                    <PstlAdr>
                        <StrtNm>999 Any Avenue</StrtNm>
                        <PstCd>10000</PstCd>
                        <TwnNm>New York</TwnNm>
                        <CtrySubDvsn>NY</CtrySubDvsn>
```

```xml
                <Ctry>US</Ctry>
            </PstlAdr>
        </Cdtr>
        <CdtrAcct>
            <Id>
                <Othr>
                    <Id>0123456789</Id>
                </Othr>
            </Id>
            <Ccy>USD</Ccy>
        </CdtrAcct>
        <RmtInf>
            <Ustrd>Testing</Ustrd>
        </RmtInf>
    </CdtTrfTxInf>
    <CdtTrfTxInf>
        <PmtId>
            <InstrId>100FED000000</InstrId>
            <EndToEndId>0000000000000065</EndToEndId>
        </PmtId>
        <Amt>
            <InstdAmt Ccy="USD">3.00</InstdAmt>
        </Amt>
        <CdtrAgt>
            <FinInstnId>
                <ClrSysMmbId>
                    <ClrSysId>
                        <Cd>USABA</Cd>
                    </ClrSysId>
                    <MmbId>123456789</MmbId>
                </ClrSysMmbId>
                <Nm>Cross River Bank NJ</Nm>
                <PstlAdr>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </FinInstnId>
        </CdtrAgt>
        <Cdtr>
            <Nm>John Doe Funding LLC</Nm>
            <PstlAdr>
                <StrtNm>999 Any Avenue</StrtNm>
                <PstCd>10000</PstCd>
                <TwnNm>New York</TwnNm>
                <CtrySubDvsn>NY</CtrySubDvsn>
                <Ctry>US</Ctry>
            </PstlAdr>
```

```xml
            </Cdtr>
            <CdtrAcct>
                <Id>
                    <Othr>
                        <Id>0123456789</Id>
                    </Othr>
                </Id>
                <Ccy>USD</Ccy>
            </CdtrAcct>
            <RmtInf>
                <Ustrd>Testing 3</Ustrd>
            </RmtInf>
        </CdtTrfTxInf>
        <CdtTrfTxInf>
            <PmtId>
                <InstrId>100FED000000</InstrId>
                <EndToEndId>0000000000000066</EndToEndId>
            </PmtId>
            <Amt>
                <InstdAmt Ccy="USD">2.01</InstdAmt>
            </Amt>
            <CdtrAgt>
                <FinInstnId>
                    <ClrSysMmbId>
                        <ClrSysId>
                            <Cd>USABA</Cd>
                        </ClrSysId>
                        <MmbId>123456789</MmbId>
                    </ClrSysMmbId>
                    <Nm>THE BANK OF NEW YORK MELLON</Nm>
                    <PstlAdr>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </FinInstnId>
            </CdtrAgt>
            <Cdtr>
                <Nm>John Doe Funding LLC</Nm>
                <PstlAdr>
                    <StrtNm>999 Any Avenue</StrtNm>
                    <PstCd>10000</PstCd>
                    <TwnNm>New York</TwnNm>
                    <CtrySubDvsn>NY</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Cdtr>
            <CdtrAcct>
```

```
                    <Id>
                        <Othr>
                            <Id>0123456789</Id>
                        </Othr>
                    </Id>
                    <Ccy>USD</Ccy>
                </CdtrAcct>
                <RmtInf>
                    <Ustrd>Testing 2</Ustrd>
                </RmtInf>
            </CdtTrfTxInf>
        </PmtInf>
    </CstmrCdtTrfInitn>
```

4.6.5.4. RTP/FedNow

> For RTP/FedNow the PAIN standard is **PAIN.001.001.03**.

Cross River will choose for you whether to use FedNow or RTP via TCH.

XML

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>DOMT11234562</MsgId>
            <CreDtTm>2020-06-29T10:24:09</CreDtTm>
            <NbOfTxs>1</NbOfTxs>
            <CtrlSum>0.01</CtrlSum>
            <InitgPty>
                <Nm>John Doe Corporation</Nm>
                <Id>
                    <OrgId>
                        <BICOrBEI>JOHNDOE99</BICOrBEI>
                    </OrgId>
                </Id>
            </InitgPty>
        </GrpHdr>
        <PmtInf>
            <PmtInfId>DOMT12345678</PmtInfId>
            <PmtMtd>TRF</PmtMtd>
            <BtchBookg>false</BtchBookg>
            <NbOfTxs>1</NbOfTxs>
            <CtrlSum>0.01</CtrlSum>
            <PmtTpInf>
                <InstrPrty>NORM</InstrPrty>
                <SvcLvl>
                    <Cd>URNS</Cd>
                </SvcLvl>
                 <LclInstrm>
                    <Prtry>RTP</Prtry>
                </LclInstrm>
            </PmtTpInf>
            <ReqdExctnDt>2020-06-29</ReqdExctnDt>
            <Dbtr>
                <Nm>John Doe Corporation</Nm>
                <PstlAdr>
                    <StrtNm>999 Battery Street, 13th Floor</StrtNm>
                    <PstCd>99999</PstCd>
                    <TwnNm>Anytown</TwnNm>
                    <CtrySubDvsn>CA</CtrySubDvsn>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </Dbtr>
            <DbtrAcct>
                <Id>
```

```xml
                <Othr>
                    <Id>0123456789</Id>
                </Othr>
            </Id>
            <Ccy>USD</Ccy>
        </DbtrAcct>
        <DbtrAgt>
            <FinInstnId>
                <ClrSysMmbId>
                    <ClrSysId>
                        <Cd>USABA</Cd>
                    </ClrSysId>
                    <MmbId>123456789</MmbId>
                </ClrSysMmbId>
                <PstlAdr>
                    <Ctry>US</Ctry>
                </PstlAdr>
            </FinInstnId>
        </DbtrAgt>
        <ChrgBr>SHAR</ChrgBr>
        <CdtTrfTxInf>
            <PmtId>
                <InstrId>234ACHC123455</InstrId>
                <EndToEndId>0000000000000011</EndToEndId>
            </PmtId>
            <Amt>
                <InstdAmt Ccy="USD">0.01</InstdAmt>
            </Amt>
            <CdtrAgt>
                <FinInstnId>
                    <ClrSysMmbId>
                        <ClrSysId>
                            <Cd>USABA</Cd>
                        </ClrSysId>
                        <MmbId>123456789</MmbId>
                    </ClrSysMmbId>
                    <Nm>Cross River Bank NJ</Nm>
                    <PstlAdr>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </FinInstnId>
            </CdtrAgt>
            <Cdtr>
                <Nm>John Doe Funding LLC</Nm>
                <PstlAdr>
                    <StrtNm>999 Any Avenue</StrtNm>
```

```xml
                        <PstCd>10000</PstCd>
                        <TwnNm>New York</TwnNm>
                        <CtrySubDvsn>NY</CtrySubDvsn>
                        <Ctry>US</Ctry>
                    </PstlAdr>
                </Cdtr>
                <CdtrAcct>
                    <Id>
                        <Othr>
                            <Id>0123456789</Id>
                        </Othr>
                    </Id>
                    <Tp>
                        <Cd>CACC</Cd>
                    </Tp>
                    <Ccy>USD</Ccy>
                </CdtrAcct>
                <RmtInf>
                    <Ustrd>Testing</Ustrd>
                </RmtInf>
            </CdtTrfTxInf>
        </PmtInf>
    </CstmrCdtTrfInitn>
</Document>
```

4.6.5.5. ACH direct debit

> For ACH Direct Debit, the PAIN standard is **PAIN.008.001.02**. This is seen in the 2nd line of the example below.

The example below is for the ACH Direct Debit CCD rail. The XML for ACH Direct Debit PPD is identical to that of ACH Direct Debit CCD, except that the value of the **Prtry** element is "PPD".

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.008.001.02">
  <CstmrDrctDbtInitn>
    <GrpHdr>
      <MsgId>DOMC10000029</MsgId>
      <CreDtTm>2020-09-03T16:32:48</CreDtTm>
      <NbOfTxs>3</NbOfTxs>
      <CtrlSum>6.00</CtrlSum>
      <InitgPty>
        <Nm>John Doe Corporation</Nm>
        <Id>
          <OrgId>
            <Othr>
              <Id>7812871377</Id>
            </Othr>
          </OrgId>
        </Id>
      </InitgPty>
    </GrpHdr>
    <PmtInf>
      <PmtInfId>DOMC10000029</PmtInfId>
      <PmtMtd>DD</PmtMtd>
      <BtchBookg>true</BtchBookg>
      <NbOfTxs>3</NbOfTxs>
      <CtrlSum>6.00</CtrlSum>
      <PmtTpInf>
        <SvcLvl>
          <Cd>NURG</Cd>
        </SvcLvl>
        <LclInstrm>
          <Prtry>CCD</Prtry>
        </LclInstrm>
      </PmtTpInf>
      <ReqdColltnDt>2020-09-03</ReqdColltnDt>
      <Cdtr>
        <Nm>Upgrade Corporation</Nm>
        <PstlAdr>
          <Ctry>US</Ctry>
          <AdrLine>275 Battery Street, 23rd Floor</AdrLine>
          <AdrLine>94111 San Francisco</AdrLine>
        </PstlAdr>
        <Id>
          <OrgId>
            <Othr>
              <Id>7812871377</Id>
```

```xml
            </Othr>
          </OrgId>
        </Id>
      </Cdtr>
      <CdtrAcct>
        <Id>
          <Othr>
            <Id>2152485385</Id>
          </Othr>
        </Id>
        <Ccy>USD</Ccy>
      </CdtrAcct>
      <CdtrAgt>
        <FinInstnId>
          <ClrSysMmbId>
            <MmbId>021214891</MmbId>
          </ClrSysMmbId>
          <PstlAdr>
            <Ctry>US</Ctry>
          </PstlAdr>
        </FinInstnId>
      </CdtrAgt>
      <DrctDbtTxInf>
        <PmtId>
          <EndToEndId>100DDEB000040</EndToEndId>
        </PmtId>
        <InstdAmt Ccy="USD">3.00</InstdAmt>
        <DbtrAgt>
          <FinInstnId>
            <ClrSysMmbId>
              <MmbId>123345678</MmbId>
            </ClrSysMmbId>
            <Nm>DUMMY</Nm>
            <PstlAdr>
              <Ctry>US</Ctry>
            </PstlAdr>
          </FinInstnId>
        </DbtrAgt>
        <Dbtr>
          <Nm>DUMMY</Nm>
          <PstlAdr>
            <Ctry>US</Ctry>
          </PstlAdr>
        </Dbtr>
        <DbtrAcct>
          <Id>
```

```xml
            <Othr>
              <Id>61651561516</Id>
            </Othr>
          </Id>
        </DbtrAcct>
        <RmtInf>
          <Ustrd>Testing</Ustrd>
        </RmtInf>
      </DrctDbtTxInf>
      <DrctDbtTxInf>
        <PmtId>
          <EndToEndId>100DDEB000042</EndToEndId>
        </PmtId>
        <InstdAmt Ccy="USD">2.00</InstdAmt>
        <DbtrAgt>
          <FinInstnId>
            <ClrSysMmbId>
              <MmbId>123345678</MmbId>
            </ClrSysMmbId>
            <Nm>DUMMY</Nm>
            <PstlAdr>
              <Ctry>US</Ctry>
            </PstlAdr>
          </FinInstnId>
        </DbtrAgt>
        <Dbtr>
          <Nm>DUMMY</Nm>
          <PstlAdr>
            <Ctry>US</Ctry>
          </PstlAdr>
        </Dbtr>
        <DbtrAcct>
          <Id>
            <Othr>
              <Id>61651561516</Id>
            </Othr>
          </Id>
        </DbtrAcct>
        <RmtInf>
          <Ustrd>Testing</Ustrd>
        </RmtInf>
      </DrctDbtTxInf>
      <DrctDbtTxInf>
        <PmtId>
          <EndToEndId>100DDEB000041</EndToEndId>
        </PmtId>
```

```xml
        <InstdAmt Ccy="USD">1.00</InstdAmt>
        <DbtrAgt>
          <FinInstnId>
            <ClrSysMmbId>
              <MmbId>123345678</MmbId>
            </ClrSysMmbId>
            <Nm>DUMMY</Nm>
            <PstlAdr>
              <Ctry>US</Ctry>
            </PstlAdr>
          </FinInstnId>
        </DbtrAgt>
        <Dbtr>
          <Nm>DUMMY</Nm>
          <PstlAdr>
            <Ctry>US</Ctry>
          </PstlAdr>
        </Dbtr>
        <DbtrAcct>
          <Id>
            <Othr>
              <Id>61651561516</Id>
            </Othr>
          </Id>
        </DbtrAcct>
        <RmtInf>
          <Ustrd>Testing</Ustrd>
        </RmtInf>
      </DrctDbtTxInf>
    </PmtInf>
  </CstmrDrctDbtInitn>
```

4.6.6. Output files

When we process your payment instructions, XML output files (PSRs) are created conforming to the ISO 20022 standard's PAIN.002.001.03 format. The output files provide information about the status of your transactions within Cross River's banking core, such as whether the transactions were accepted or rejected.

Instead of an acknowledgment file followed by a PSR later on, Cross River sends you the PSR immediately.

See **Status and error codes** for more information.

File naming convention

The output file PSRs conform to the following naming convention:

{InputFileName}.{date}.{time}-{number}

where "number" is the ordinal number of file from the specified date and time, and is used to differentiate among multiple possible files that were generated at that date and time. If only one file was generated at that date and time, the value of "number" will be "1".

How an output file is built

The root element in the Output file is **CstmrPmtStsRpt** (Customer Payment Status Report) and it contains the following 3 building blocks:

- **GrpHdr** that contains the file metadata.
- **OrgnlGrpInfAndSts** contains message identifiers based on the original input file and includes a status for the group.
- **OrgnlPmtInfAndSts** contains elements referencing the original transactions. This is an optional section that may appear multiple times in the file. It can contain an individual status for the original instruction, as well as elements from your original transaction file.

Here's an example of the file structure:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Document xmlns='urn:iso:std:iso:20022:tech:xsd:pain.002.001.03' xmlns:xsi='h
  <CstmrPmtStsRpt>
    <GrpHdr>
      ...
    </GrpHdr>
    <OrgnlGrpInfAndSts>
      ...
    </OrgnlGrpInfAndSts>
    <OrgnlPmtInfAndSts>
      ...
      <TxInfAndSts>
        ...
      </TxInfAndSts>
    </OrgnlPmtInfAndSts>
  </CstmrPmtStsRpt>
</Document>
```

Group header

The **GrpHdr** block is a set of characteristics shared by all individual transactions included in the message. This building block is present once per output file. The group header includes the following elements:

```xml
<GrpHdr>
  <MsgId>11000000000a0a0</MsgId>
  <CreDtTm>2016-05-10T15:20:06.719Z</CreDtTm>
  <InitgPty>
    ...
  </InitgPty>
</GrpHdr>
```

- **MsgId**: Message ID. A unique ID that we generate for the file. It is unique per instructed party for a pre-agreed period. Best practice: Use a new for each file.**

- **CreDtTm**: Created Date and Time. This is the date and time at which the output file was created.
- **InitgPty**: Initiating Party. This element is used to indicate the party that initiated the file. With the output files this will always be CR. The **InitgPty** block appears as follows:

XML

```xml
<InitgPty>
  <Nm>John Doe Corporation</Nm>
  <Id><OrgId> <BICOrBEI>JOHNDOE99</BICOrBEI>
  </OrgId></Id>
</InitgPty>
```

- **Nm**: Name used to identify the initiating party.
- **Id**: The unique identifier for the initiating party.
- **OrgId**: Organization Identification, containing the BICOrBEI code.
- **BICOrBEI**: Business Identification Code. A unique and unambiguous identifier of the initiating party, which can be an organization or an individual person.

Original group information and status

Various information can be included for the group in the **OrgnlGrpInfAndSts** block. The information included will depend on the input file and the status of the original transaction.

- **OrgnlMsgId** is the message identification number from the original input file.
- **OrgnlMsgNmId** indicates the original message type, and will either be pain.001.001.03 for payments (USD) or pain.008.001.02 for direct debits (USD).
- **OrgnlNbOfTxs** shows the number of transactions in the original input file.
- **OrgnlCtrlSum** displays the control sum from the original input file.
- **GrpSts** indicates the status for a group of transactions. See here for details about the different statuses.
- **AddtlInf** shows details for the status in the case of a rejected file. The rejection reason can be due to an issue with the original XML file, as well as an issue arising during internal processing by CR.

Original transaction file accepted

The following example of the **OrgnlGrpInfAndSts** block shows that the original input transaction file was accepted. You can see this in the **GrpSts** element, where "ACTC" indicates the file was accepted.

XML

```xml
<OrgnlGrpInfAndSts>
    <OrgnlMsgId>ABCDEFG090301</OrgnlMsgId>
    <OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
    <OrgnlNbOfTxs>1</OrgnlNbOfTxs>
    <OrgnlCtrlSum>42720.80</OrgnlCtrlSum>
    <GrpSts>ACTC</GrpSts>
</OrgnlGrpInfAndSts>
```

Original transaction file rejected

The following example of the **OrgnlGrpInfAndSts** block shows that the original input transaction file was rejected. You can see this in the **GrpSts** element, where "RJCT" indicates the file was rejected. In this case, the output file will include the **StsRsnInf** element, which gives information about why the initial transaction was rejected. The reason will always be shown in the AddtlInf element.

> If your original transaction file is rejected, you should correct any errors and resend it to CR.

In this example, the reason the file was rejected is shown as “Number of transactions in file does not match with control record: Expected 1, but calculated 2”. This means that there is a mismatch between the **OrgnlCtrlSum** provided in the original file and the actual number of transactions provided there.

```xml
<OrgnlGrpInfAndSts>
  <OrgnlMsgId>ABCDEFG090301</OrgnlMsgId>
  <OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
  <OrgnlCreDtTm>2015-08-10T10:48:47</OrgnlCreDtTm>
  <GrpSts>RJCT</GrpSts>
  <StsRsnInf>
    <Rsn>
      <Cd>NARR</Cd>
    </Rsn>
    <AddtlInf>Number of transactions in file does not match with control reco
  </StsRsnInf>
</OrgnlGrpInfAndSts>
```

Original payment information and status

OrgnlPmtInfAndSts is a third and optional block, which contains elements referencing the original transactions. It corresponds to the batch level in the original file (specifically the **PmtInf** block), and can also include individual statuses for the original instructions.

```xml
<OrgnlPmtInfAndSts>
      <OrgnlPmtInfId>ABCDEFG00065</OrgnlPmtInfId>
      <OrgnlNbOfTxs>1</OrgnlNbOfTxs>
      <OrgnlCtrlSum>42720.80</OrgnlCtrlSum>
      <PmtInfSts>ACTC</PmtInfSts>
      <TxInfAndSts>
        ...
      </TxInfAndSts>
    </OrgnlPmtInfAndSts>
```

- **OrgnlPmtInfId** is a unique identifier, assigned by the original initiating party. It's used to identify the original payment information block within the output file.
- **OrgnlNbOfTxs** indicates the number of payments in the original payment information block.
- **OrgnlCtrlSum** shows the sum total of all transactions included in the original payment information block.

- **PmtInfSts** indicates the status of the payment information block as follows:

Payment Info Status	Meaning
ACTC	The batch was validated and authenticated successfully
RCVD	The batch is being checked
RJCT	The batch was rejected
PDNG	One or more transactions in the batch is pending
PART	The transaction batch was partially accepted and partially rejected. This means that the batch includes one or more rejected transactions as well as one or more transactions that have been accepted. In the output file generated by, you'll be able to see which transactions were rejected so you can make the necessary changes and resend to CR.

> If your original transaction file is rejected, you should correct any errors and resend to CR.

Transaction-level information

The output file can include information about individual instructions in the original file in a building block called **TxInfAndSts**. This will be included as part of the **OrgnlPmtInfAndSts** building block. The transaction-level information includes the following:

- **OrgnlInstrId** contains the value of the Instruction Identification (**InstrId**) element provided in the input file. This element won't be present if an **InstrId** value wasn't provided in the input file.
- **OrgnlEndToEndId** contains the value of the **EndtoEndId** element provided in the input file.
- **TxSts** indicates the transaction status using one of the following codes:

Status	Meaning
ACPT	The transaction has been accepted by the external payment processor for processing
RJCT	The transaction was rejected
ASCS	The transaction has been completed and the settlement has been confirmed on the debtor's account
ACSP	The transaction was accepted and has been moved on for processing
PDNG	The transaction is pending and requires further processing

In addition, the **TxInfAndSts** block contains the following elements:

- **OrgnlTxRef**: Contains information on the original transaction in its child elements, such as the transaction amount and the transaction's requested execution date.
- **Amt**: Contains the amount of the original transaction in a child element.
- **InstdAmt**: Contains the amount of the original transaction in the indicated currency.
- **ReqdExtnDt**: Requested Execution Date. The date on which the originator's account was to be debited.

If the status is shown as RJCT (rejected) an additional element, **StsRsnInf**, will be included in the **TxInfAndSts** block. This will provide information on why the transaction was rejected.

Here’s an example of the **TxInfAndSts** building block for when a transaction has succeeded:

```xml
<TxInfAndSts>
        <OrgnlInstrId>abcdefg000000</OrgnlInstrId>
        <OrgnlEndToEndId>0000000000001011</OrgnlEndToEndId>
        <TxSts>ACSC</TxSts>
        <OrgnlTxRef>
          <Amt>
            <InstdAmt Ccy="USD">42720.80</InstdAmt>
          </Amt>
          <ReqdExctnDt>2016-05-10</ReqdExctnDt>
        </OrgnlTxRef>
</TxInfAndSts>
```

Below is an example the **TXInfAndSts** building block for when a transaction has been rejected:

```xml
<TxInfAndSts>
        <OrgnlInstrId>abcdefg000000</OrgnlInstrId>
        <OrgnlEndToEndId>0000000000001011</OrgnlEndToEndId>
        <TxSts>RJCT</TxSts>
            <StsRsnInf>
            <Rsn>
                <Cd>NARR</Cd>
            </Rsn>
            <AddtlInf>The account 1234567890 is not setup in Unicorn for Clie
            </StsRsnInf>
</TxInfAndSts>
```

The **StsRsnInf** block, which appears inside the **TxInfAndSts** block in the case of a rejected transaction, contains information on why the transaction was rejected, and contains the following elements:

- **Rsn**: Contains the reason code in a child element.
- **Cd**: Code. Currently, this code will always have the value of NARR, standing for "Narrative". The actual narrative text appears in the **AddtlInf** element.

- **AddtlInf**: Shows details for the status in the case of a rejected file. The rejection reason can be due to an issue with the original XML file, as well as an issue arising during internal processing by CR.

Full sample output files

Sample successful transaction

The examples below shows full output files generated when we have successfully processed the transaction file, for both the pain.001.001.03 and pain.008.001.002 formats. These examples include the Group Header, Original Information And Status, and Original Payment Information And Status blocks.

From pain.001.001.03 input file

The following is an example output file corresponding to a **pain.001.001.03 input file**, generated when the processing of the transaction file has succeeded.

XML

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Document Xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03" xmlns:xsi="h
  <CstmrPmtStsRpt>
    <GrpHdr>
      <MsgId>abcdefg0000000</MsgId>
      <CreDtTm>2020-09-15T15:58:04.637Z</CreDtTm>
      <InitgPty>
        <Nm>John Doe Corporation</Nm>
        <Id>
          <OrgId>
            <BICOrBEI>JOHNDOE99</BICOrBEI>
          </OrgId>
        </Id>
      </InitgPty>
    </GrpHdr>
    <OrgnlGrpInfAndSts>
      <OrgnlMsgId>AMBCD10000000</OrgnlMsgId>
      <OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
      <OrgnlNbOfTxs>3</OrgnlNbOfTxs>
      <OrgnlCtrlSum>5.0100</OrgnlCtrlSum>
      <GrpSts>ACTC</GrpSts>
    </OrgnlGrpInfAndSts>
    <OrgnlPmtInfAndSts>
      <OrgnlPmtInfId>ABCD10000000</OrgnlPmtInfId>
      <OrgnlNbOfTxs>3</OrgnlNbOfTxs>
      <OrgnlCtrlSum>5.0100</OrgnlCtrlSum>
      <PmtInfSts>ACTC</PmtInfSts>
      <TxInfAndSts>
        <OrgnlInstrId>100ABCD012345</OrgnlInstrId>
        <OrgnlEndToEndId>0000000000000108</OrgnlEndToEndId>
        <ClrSysRef>46456456</ClrSysRef>
        <TxSts>ACSP</TxSts>
      </TxInfAndSts>
    </OrgnlPmtInfAndSts>
  </CstmrPmtStsRpt>
</Document>
```

From pain.008.001.02 input file

The following is an example output file corresponding to a **pain.008.001.02 input file**, generated when the processing of the transaction file has succeeded.

XML

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Document Xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03" xmlns:xsi="h
  <CstmrPmtStsRpt>
    <GrpHdr>
      <MsgId>abcdefg0123456789</MsgId>
      <CreDtTm>2020-11-02T11:30:11.055Z</CreDtTm>
      <InitgPty>
        <Nm>John Doe Corporation</Nm>
        <Id>
          <OrgId>
            <BICOrBEI>JOHNDOE99</BICOrBEI>
          </OrgId>
        </Id>
      </InitgPty>
    </GrpHdr>
    <OrgnlGrpInfAndSts>
      <OrgnlMsgId>DD1</OrgnlMsgId>
      <OrgnlMsgNmId>pain.008.001.02</OrgnlMsgNmId>
      <OrgnlNbOfTxs>3</OrgnlNbOfTxs>
      <OrgnlCtrlSum>6.0000</OrgnlCtrlSum>
      <GrpSts>ACTC</GrpSts>
    </OrgnlGrpInfAndSts>
    <OrgnlPmtInfAndSts>
      <OrgnlPmtInfId>DD1-1</OrgnlPmtInfId>
      <OrgnlNbOfTxs>3</OrgnlNbOfTxs>
      <OrgnlCtrlSum>6.0000</OrgnlCtrlSum>
      <PmtInfSts>ACTC</PmtInfSts>
      <TxInfAndSts>
        <OrgnlEndToEndId>DD1-1</OrgnlEndToEndId>
        <ClrSysRef>012345678901234</ClrSysRef>
        <TxSts>ACSP</TxSts>
      </TxInfAndSts>
      <TxInfAndSts>
        <OrgnlEndToEndId>DD1-2</OrgnlEndToEndId>
        <ClrSysRef>012345678901234</ClrSysRef>
        <TxSts>ACSP</TxSts>
      </TxInfAndSts>
      <TxInfAndSts>
        <OrgnlEndToEndId>DD1-3</OrgnlEndToEndId>
        <ClrSysRef>012345678901234</ClrSysRef>
        <TxSts>ACSP</TxSts>
      </TxInfAndSts>
    </OrgnlPmtInfAndSts>
```

Sample rejected transaction

The examples below show full output files generated when the transaction file has been rejected.

> In such output files, the error message information appears in the tag.

Rejected file (batch level error)

The following example shows an output file generated when the transaction file has been rejected on the batch level, for an original file that came in **pain.008.001.02** format.

XML

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Document Xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03" xmlns:xsi="h
  <CstmrPmtStsRpt>
    <GrpHdr>
      <MsgId>abcdefg012345689</MsgId>
      <CreDtTm>2020-11-02T07:56:06.228Z</CreDtTm>
      <InitgPty>
        <Nm>John Doe Corporation</Nm>
        <Id>
          <OrgId>
            <BICOrBEI>JOHNDOE99</BICOrBEI>
          </OrgId>
        </Id>
      </InitgPty>
    </GrpHdr>
    <OrgnlGrpInfAndSts>
      <OrgnlMsgNmId>pain.008.001.02</OrgnlMsgNmId>
      <OrgnlNbOfTxs>0</OrgnlNbOfTxs>
      <OrgnlCtrlSum>0.0000</OrgnlCtrlSum>
      <GrpSts>RJCT</GrpSts>
      <StsRsnInf>
        <Rsn>
          <Cd>NARR</Cd>
        </Rsn>
        <AddtlInf><![CDATA[Error parsing file examplefile.xml: The 'MsgId' st
      </StsRsnInf>
    </OrgnlGrpInfAndSts>
  </CstmrPmtStsRpt>
</Document>
```

Rejected file (transaction level error)

The following example shows an output file generated when the transaction file has been rejected on the transaction level, for an original file that came in **pain.001.001.003** format.

XML

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Document Xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03" xmlns:xsi="h
  <CstmrPmtStsRpt>
    <GrpHdr>
      <MsgId>abcdefg0123456789</MsgId>
      <CreDtTm>2020-10-26T11:49:33.620Z</CreDtTm>
      <InitgPty>
        <Nm>John Doe Corporation</Nm>
        <Id>
          <OrgId>
            <BICOrBEI>JOHNDOE99</BICOrBEI>
          </OrgId>
        </Id>
      </InitgPty>
    </GrpHdr>
    <OrgnlGrpInfAndSts>
      <OrgnlMsgId>missing2</OrgnlMsgId>
      <OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
      <OrgnlNbOfTxs>3</OrgnlNbOfTxs>
      <OrgnlCtrlSum>5.0400</OrgnlCtrlSum>
      <GrpSts>ACTC</GrpSts>
    </OrgnlGrpInfAndSts>
    <OrgnlPmtInfAndSts>
      <OrgnlPmtInfId>missing2</OrgnlPmtInfId>
      <OrgnlNbOfTxs>3</OrgnlNbOfTxs>
      <OrgnlCtrlSum>5.0400</OrgnlCtrlSum>
      <PmtInfSts>RJCT</PmtInfSts>
      <TxInfAndSts>
        <OrgnlInstrId>example01234567</OrgnlInstrId>
        <OrgnlEndToEndId>sanity-10-1-12-1-21</OrgnlEndToEndId>
        <TxSts>RJCT</TxSts>
        <StsRsnInf>
          <Rsn>
            <Cd>NARR</Cd>
          </Rsn>
          <AddtlInf><![CDATA[Instruction ID example01234567 already exist]]><
        </StsRsnInf>
      </TxInfAndSts>
      <TxInfAndSts>
        <OrgnlInstrId>example01234567</OrgnlInstrId>
        <OrgnlEndToEndId>example01234567</OrgnlEndToEndId>
        <TxSts>RJCT</TxSts>
        <StsRsnInf>
          <Rsn>
            <Cd>NARR</Cd>
```

```
            </Rsn>
            <AddtlInf><![CDATA[Instruction ID example01234567 already exist]]><
          </StsRsnInf>
        </TxInfAndSts>
        <TxInfAndSts>
          <OrgnlInstrId>example012345678</OrgnlInstrId>
          <OrgnlEndToEndId>example012345678</OrgnlEndToEndId>
          <TxSts>RJCT</TxSts>
          <StsRsnInf>
            <Rsn>
              <Cd>NARR</Cd>
            </Rsn>
            <AddtlInf><![CDATA[Instruction ID example012345678 already exist]]>
          </StsRsnInf>
        </TxInfAndSts>
      </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
  </Document>
```

4.6.7. Validation

Initial validation on the submitted input files ensures that we received the relevant information needed to initiate the payment requests and report on them.

Additional validations will be executed by Cross Rivers' payment system before processing the transactions.

In both cases, if a validation error is found or we cannot process the request for any other reason, a report will be generated and returned to you describing what is wrong with the input file so that you can re-submit it.

Considerations

The following are some validation considerations:

- The file must contain valid XML only.
- The XML must follow the ISO 20022 **PAIN.001.001.03** or **PAIN.008.001.02** format.
- The sum of the transaction amounts in the file must not exceed the File Level Limit configured per your agreement with Cross River.
- You can only use the rails that have been agreed upon with us. At any point, you can request to update the setup for the rails you are using.
- The number of transactions in the file must equal the Control number of transactions specified in the file's Group Header. Each batch's Payment Information NbOfTxs field must equal the number of transactions it contains. Refer to **Prepare input XML file** .
- The sum of the transaction amounts must equal the Control amount specified in the Group Header or Payment Information CtrlSum field. Refer to **Prepare input XML file**.
- The payment ID (PmtId) is validated so doubles are not sent - if a duplicate payment ID is found, **the instruction will be ignored**.
- The account number used must have been registered at the outset.
- The Requested Execution Date must not be in the future.
- The Creditor and Debtor account currency must be in USD.

4.6.8. Status and error codes

Processing of your **PAIN.001.001.03** and **PAIN.008.001.02** input files creates PAIN.002.001.03 XML output files containing status and error information. We provide status at the following levels when processing the files received from a customer:

- **Group**: Header information shown once in the file that is common to all the transactions in that file.
- **Batch**: A block of either one or more transactions.
- **Transaction**: A block containing a single transaction.

Group header

The group header building block in the XML file will show the file-level status in the **GrpSts** tag as follows:

Status	Meaning
ACTC	The file was validated and authenticated successfully
RCVD	The file has been received but is still being validated
RJCT	The file was rejected

If a file is rejected, additional information is shown in the **StsRsnInf** tag. The Cd "NARR" value indicates that there is a reason the file was rejected, with the reason description appearing in the **AddtlInf** tag as shown in this example:

```xml
<GrpSts>RJCT</GrpSts>
  <StsRsnInf>
    <Rsn>
      <Cd>NARR</Cd>
    </Rsn>
    <AddtlInf>Number of transactions in file does not match with control reco
  </StsRsnInf>
```

Here are some examples of reasons that can appear in the Additional information:

- The number of transactions in the file doesn't match the number in the control record.
- The sum of all transactions in the file doesn't match the number in the control record.
- The total in the file exceeds the limit for the customer. In this case, you should contact the CR Client Integration team.
- A validation error was found in the file.

Batch level

The batch building block shows batch status as follows:

Status	Meaning
ACTC	The batch was validated and authenticated successfully
RCVD	The batch is being checked
RJCT	The batch was rejected
PNDG	One or more transactions in the batch is pending
PART	The transaction batch was partially accepted and partially rejected. This means that the batch includes one or more rejected transactions as well as one or more transactions that have been accepted. In the output file generated, you'll be able to see which transactions were rejected so you can make the necessary changes and resend to CR.

If a batch is rejected, additional information is shown in the **StsRsnInf** tag. The Cd NARR indicates the reason the file was rejected, with the description appearing in the **AddtlInf** tag. Here are some examples of what might appear in this tag:

- The number of transactions doesn't match the number in the control record.
- The control sums don't correspond with the total calculated.
- The requested execution date is later than the current date and time.
- The creditor's account currency isn't in USD.
- The debtor's account currency isn't in USD.

Transaction level

Statuses at the transaction level can be one of the following:

Status	Meaning
ACPT	The transaction has been accepted by the external payment processor for processing
RJCT	The transaction was rejected
ACSC	The transaction has been completed and the settlement has been confirmed on the debtor's account
ACSP	The transaction was accepted and has been moved on for processing
PDNG	The transaction is pending and requires further processing

If the transaction is rejected, additional information is shown in the **StsRsnInf** tag. The **Cd** tag will show either as "NARR" for narrative or "ARET" for a return, with a reason description appearing in the **AddtlInf** tag. Here are some examples of what might appear in this tag:

- The customer isn't enabled for this rail. In this case, you should contact the CR Client Integration team.
- The account is not set up for this customer. In this case, you should contact the CR Client Integration team.

- In the case of a return, cancellation, or Notice of Change (NOC), a reason code and trace number will be provided here.
- In the case of a wire cancellation, a message will tell you to please contact your RM for more information on this canceled wire.

4.7. Checks

Checks APIs

Cross River delivers both check writing (withdrawals) and check deposits using MDC. Cross River works with lockbox service providers to process paper check deposits.

What Cross River offers

In Cross River, checks have two different flows:

- **Deposit.** In the deposit flow, you deposit a check into your Cross River account. In Cross River this is referred to as an *outbound forward* item because the check needs to be sent (*out*) to the Federal Reserve and presented to the drawee bank (paying bank) for payment.
- **Withdrawal.** In the withdrawal flow, you write a check on your Cross River account to pay someone. In Cross River this is referred to as an *inbound forward* item because the Federal Reserve sends the check (*in*) to Cross River for payment and attempts to clear the check on your account. Your check *clears* when the transfer of funds succeeds.

Find out more

- **Transaction roles**: Basic roles describing the participants in a check transaction.
- **Availability policies**: Defines how quickly funds can be made available to the payee.
- **Positive pay**: Cross River's Positive Pay feature lets you or your customer make sure that only legitimate checks get deposited in their accounts.
- **Stop payments**: If the payer has passed a check that the payer FI has not yet processed, they can put a stop payment on the check.

Tutorials

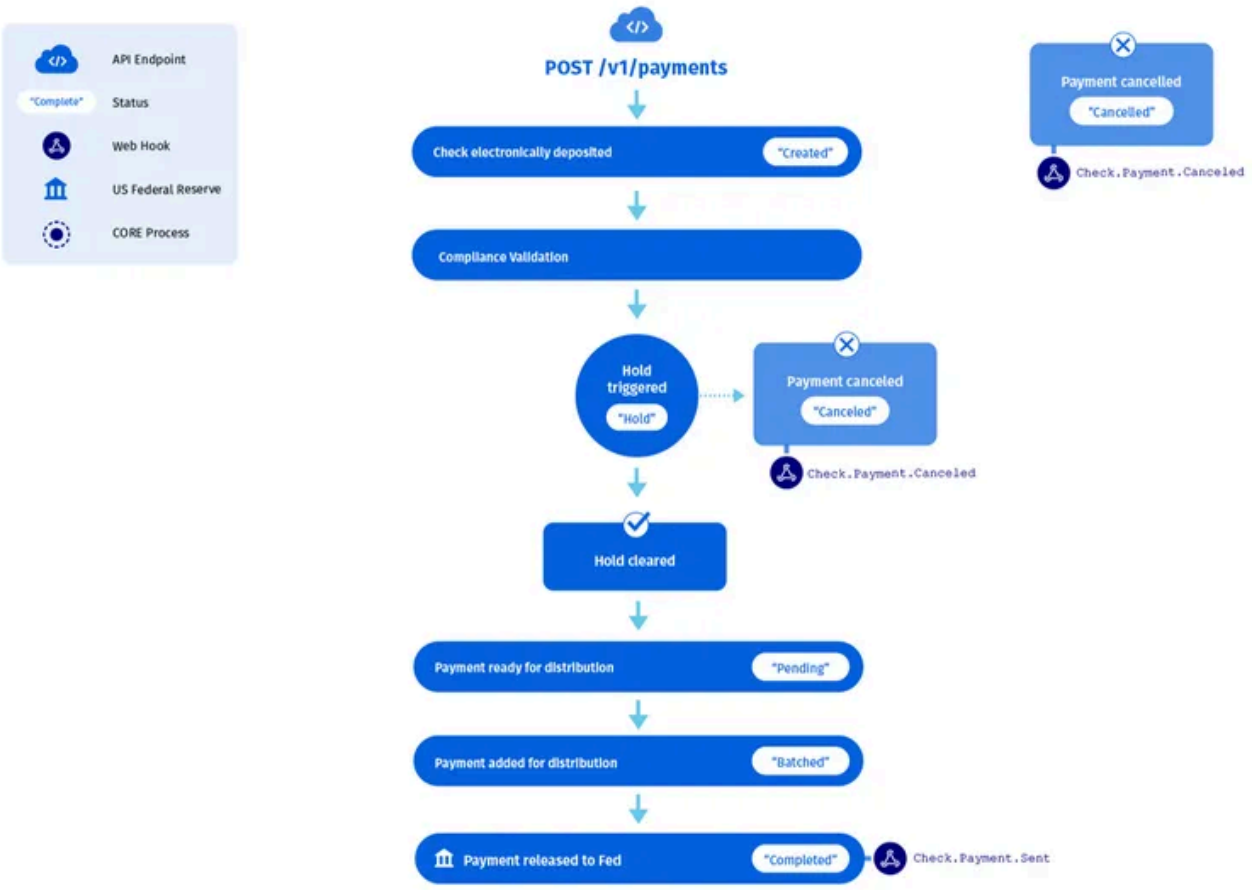
How to deposit a check

4.7.1. Send and receive

Outbound forward

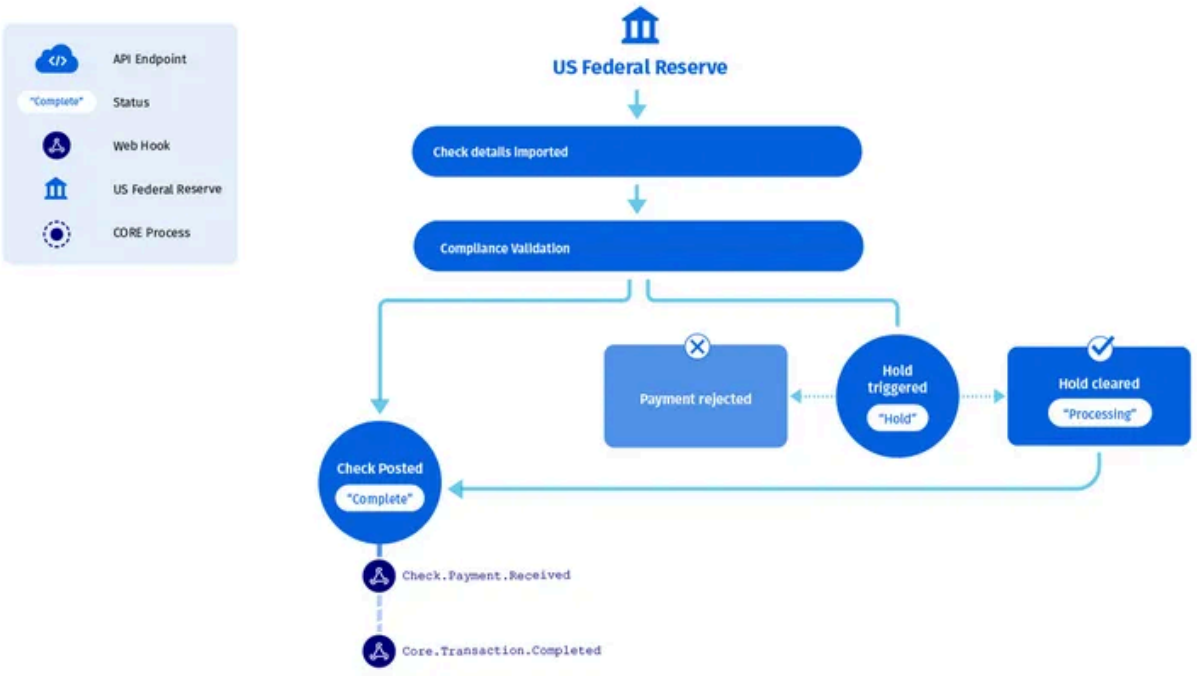
The following diagram shows the status and webhook flows for *outbound forward* check payments.

> See our tutorial to learn how to **deposit a check**.



Inbound forward

The following diagram shows the status and webhook flows for *inbound forward* check payments:



Inbound/outbound return

Whether deposited or used as a payment, a check might be returned for a number of reasons.

- A returned check presented to Cross River by the Federal Reserve is considered an *inbound return*.
- A check returned by Cross River is considered an *outbound return.*

4.7.2. Transaction roles

The following are the basic roles describing the participants in a check transaction.

Role Name	Description
Issuing bank	The bank that issued the check (usually as part of a checkbook) to the account holder
Check payer/payer	The account holder writing and passing the check
The Federal Reserve/The Fed	The central bank of the United States
Check payee/Payee	The account holder depositing the check and receiving its funds
Depositing bank	The check payee bank. The check is deposited into the payee's account at the depositing bank.
Check payment services	A range of companies that offer a variety of automated payment services, including the option to print and mail a check on behalf of an account holder.

4.7.3. Availability policies

Availability policies define how quickly funds can be made available to the payee.

Availability of funds from a deposited check is based on rules specifically mandated by the Federal Reserve under Regulation CC (Reg CC). Reg CC defines how long a bank can hold funds based on various **types** of deposits. It's important to understand check processing and how Regulation CC (12 CFR 229) laws effect availability of funds.

Deposit a check

When you **deposit a check** with the API, the response shows the assigned availability policy in the `policy` field and the payment schedule in the `schedule` field. For example, in the response below, the check is assigned the *NewAccount* availability policy along with the schedule. The schedule shows the amount of funds from the check deposit that will be available in the customer's COS account each day starting with the first day (Day 1), which is defined as the value in the `processedAt` field. In this example, all the funds (**100** or 1.00 USD) are available on the third day.

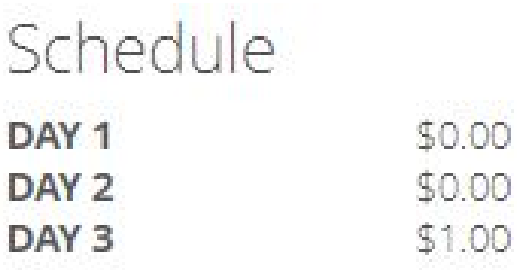
COS defines the availability policy of any given deposit when the deposit is successfully received. COS assigns the policy of the deposit based on that business day's deposits to that specific account.

You can see the `schedule` field and its array in the deposit a check response starting in row 31 below.

JSON

```
{
  "id": "9a44fdbf-89e2-4300-8f05-ad9501439bb1",
  "accountNumber": "2193590144",
  "referenceId": "C2436F698K0D",
  "paymentType": "Forward",
  "checkType": "Standard",
  "direction": "Outbound",
  "status": "Completed",
  "source": "Api",
  "posting": "Posted",
  "postingCode": "OK",
  "coreTransactionId": "21ba5fb0-e609-4560-a36f-ad9501439bb1",
  "memoPostId": "dc20e619-c5f7-4890-b3c8-ad9501439bb1",
  "originalPaymentId": "9a44fdbf-89e2-4300-8f05-ad9501439bb1",
  "customerId": "700f0d35-9940-4132-8608-ad89013927a9",
  "payerRoutingNumber": "021200339",
  "payerAccountNumber": "123456789",
  "payeeName": "",
  "checkNumber": "1237",
  "bofdRoutingNumber": "021214891",
  "sequenceNumber": "1353885824",
  "amount": 100,
  "currency": "usd",
  "micr": "d021200339dc12345678c1237",
  "recognizedAmount": 100,
  "iqaPassed": true,
  "hasFrontImage": true,
  "hasBackImage": true,
  "isRedeposit": false,
  "policy": "NewAccount",
  "schedule": [
    0,
    0,
    100
  ],
  "createdAt": "2021-08-31T15:38:13.28-04:00",
  "rejectedAt": "2021-08-31T15:38:17.527-04:00",
  "wasReturned": false,
  "purpose": "ENTERED BY #60C367E26DD36A0068580230#",
  "depositBusinessDate": "210831",
  "productId": "d5dc52bb-df80-4a5d-a5a8-ad89013844bd",
  "partnerId": "ede1a60d-3d51-47e8-9a9b-ad8901381f9e",
  "lastModifiedAt": "2021-08-31T15:38:17.5280148-04:00"
}
```

If you look at the deposit in COS Explorer, the same schedule would appear as:



Using the deposit scenario above, this means that $1.00 will be available in the account ending in 0144 on Thursday, September 2nd. Below is a more detailed breakdown of the schedule depicted in the payment details above:

Day #	Date	Amount Available
1	Tuesday, August 31, 2021	$0.00
2	Wednesday, September 1, 2021	$0.00
3	Thursday, September 2, 2021	$1.00

4.7.4. Positive pay

Cross River's Positive Pay feature lets you or your customer make sure that only legitimate checks get deposited in their accounts. Either you or the account holder define which specific checks will be accepted. This helps prevent check fraud.

You can think of Positive Pay as the opposite of **Stop Pay**. In a sense you're providing an allowed list of authorized checks for deposit. Your or your customer provide the check number and amount for authorized checks. COS takes this Positive Pay information and matches it against *inbound forward* items presented by the Fed. If a match is found the check is paid. Otherwise the check is rejected.

Positive Pay statuses appear in the `status` field in the **positive pay responses** API, and include:

Authorized	**The check has been authorized for positive pay**
Paid	An authorized check has been presented by the Federal Reserve and paid by the Bank
Revoked	The check's positive pay authorization has been revoked

In regular check calls Positive Pay statuses appear in the `positivePayResult` field, and include an additional value, **Disabled**, if Positive Pay is not configured for the check product.

> Your product must be configured by Cross River in order to use the Positive Pay feature. Once Positive Pay is enabled, any checks presented by the Fed that are not authorized will automatically be rejected and returned to the depositing bank.

> **IMPORTANT**
> The date time within the `expiresAt` field of the response is ignored. If a check matching that positive pay criteria is received *after* the date in the `expiresAt` field it will be automatically rejected and returned to the depositing bank.

Positive Pay API endpoints include:

- **Authorize a check**
- **Revoke authorization**

4.7.5. Stop payments

If the payer has passed a check that the payer FI has not yet processed, they can put a stop payment on the check.

Once a check has been processed by the payer FI, a stop payment cannot be made. Instead, a dispute must be filed by the payer.

If the payee deposits the check after the check has had a stop payment put on it, the payee FI generally charges the payee a penalty fee. The payer bank will likely also charge for the stop payment service.

Usually the check amount, check number and account number is included the stop payment instructions.

In addition, whether deposited or used as a payment, a check might be returned for a number of reasons.

- A returned check presented to Cross River by the Federal Reserve is considered an *inbound return*.
- A check returned by Cross River is considered an *outbound return*

For further guidance, please refer to: **Reg E 1005.10 Preauthorized Transfer (c) Customer's Right to Stop payment** section in the following link:

https://www.consumerfinance.gov/rules-policy/regulations/1005/10/#a-1

5. Lending

Lending APIs

As a marketplace lending (MPL) partner with Cross River you can access Lending functions via an extensive list of API calls. Loans are funded after the review and fulfillment of documentation and compliance requirements.

What Cross River offers

- **Application preapproval**: Preapproval APIs allow you to vaidate an application is meeting Cross River requirements as part of the application flow.
- **Loan origination and funding**: The Arix engine validates all loans and allows you the flexibility to control how the loan is funded. You can enter instructions and control the rails, and the order in which to attempt to fund over those rails. You can also elect to fund different parts of the loan over different rails, and supply fallback options if rails fail.
- **Loan seasoning, sale or retention**: After funding, every loan enters a seasoning period. Post-seasoning, you or your investors will purhcase the loans or retain it on Cross River's books (flexible warehousing).
- **Loan data exchange**: MPL submits servicing data on a daily basis. This powers Cross River warehousing and long term oversight process.

Find out more

- **Loan process**
- **New in lending**

5.1. Loan process

Cross River's Lending platform consists of several interconnected modules:

- **Oversight** for policies and rules
- **Preapproval** for applications
- **Arix-origination** for issuing loans
- **Manual review** - for working the exception queue detected by **MPLLifeCycle**
- **Arix-funding** - for making payments
- **Selling** - for purchasing loans
- **Contracts** - for viewing configuration
- **Hooks** - for listening to events

On a typical loan, the entire process takes around 10 seconds from origination to funding and another 5 minutes for seasoning to kick off.

1. Partner submits loan application for preapproval

The application can be submitted regardless of its approval status. If the status changes, the MPL can resubmit to provide its latest decision. This process aligns with Fair Lending laws, ensuring all applications are reviewed uniformly.

The Marketplace Lending Partner (MPL) submits a loan application to the **Preapproval** API to provide the bank with its final decision for each application.

2. Partner submits loan payload to Arix

This step initiates the loan creation process within Arix. Arix assigns a unique GUID to the loan and sets its initial status to **Received**.

Upon approval, the MPL sends the complete loan payload to Arix using the **POST /Loan API call** for:

- Consumer and business loans

- Secured and unsecured
- Term loans or revolving lines of credit

3. Partner uploads required loan documents

Arix automatically reviews the submitted documents to ensure completeness. Once all required documents are received, the loan status updates to **Docs Complete**.

The MPL uploads all necessary loan documents to Arix using the **POST /Loan/{Id}/Attachments API** call.

4. MPLLifecycle performs compliance checks

MPLLifeCycle automatically conducts compliance checks based on predefined rules from **Oversight**.

> This step is where Cross River ensures the loan details are complete and accurate

If the loan passes all compliance rules, the status updates to **Passed Compliance**. If any rule fails, the status changes to **ComplianceFailed**, and detailed reasons are provided via the **Loan Details API**. The MPL must address these issues before proceeding. Towards that end, the MPL would use the **Manual Review** module to communicate with Cross River teams, upload additional documents or correct the loan details.

5. MPLLifeCycle approves the loan

After successful compliance checks, the loan status updates to **Approved** indicating readiness for funding.

6. Arix initiates loan funding

The loan status changes to **InFunding** during this process. Arix supports various funding rails, including ACH, Same-Day ACH, RTP®, FedNow®, RPPS, Check, Internal Transfer, Wire and Push to Card. The platform allows setting fallback options if a chosen rail fails. Once funds are successfully sent and received, the loan status updates to **Funded**. Arix

manages the full lifecycle of Funding, including Rejects, Returns, pulling funds back and Cancellations.

Learn more

7. Selling kicks off loan seasoning period

After funding, the loan enters a seasoning period that is configured to your Capital Markets and/or Investors needs.

Learn more

8. Loan sale or retention decision

Post-seasoning, you or your investor will purchase the loans using **Selling**. Typical seasoning is for 3 business days, but Cross River offers warehousing from 30-120 days which you will manage self service. Cross River and you may choose to have Cross River retain loans on the bank's balance sheet.

Learn more

9. Servicing data submission and reconciliation

MPL submits servicing data on a daily basis. This is used to monitor loans on Cross River's books.

Learn more

5.2. Loan funding

The Arix funding engine allows you the flexibility to control how the loan is funded. You can enter instructions and control the rails, and the order in which to attempt to fund over those rails. You can also elect to fund different parts of the loan over different rails, and supply fallback options if rails fail.

After the loan is approved for funding, Arix initiates funding. The loan passes to the InFunding status and funds are sent using rails.

There are several **rail types** you can use. Depending on the rail type, funding results might be immediate or take up to a few days.

Based on how the funding progresses, the loan can be in one of these statuses:

- `InFunding` leads to `Funded` or `NotFullyFunded`
- `Returned`

Status: Infunding/ Funded

1. After the loan is approved, funding starts automatically. Arix tries the **Priority 1** rail first to start funding.

Status: InFunding

2. When all the rails are successfully funded, the status of the loan changes to funded
Status: Funded

3. After the loan has funded, the loan seasoning period begins, which typically occurs on business day 2 or 3 from the funding date.

> **If no amount is specified in the request, the funding engine attempts to fund the full loan amount.**

Status: NotFullyFunded

If the loan doesn't fully fund from the first rail (priority 1), the engine looks for the priority 2 rail.
A priority 2 rail can be:

- A rail that was originally designated as a priority 2 (to follow a priority 1 rail)
- The next priority 1 rail

The funding engine continues to process rails until there are no rails left to process.

If after all the rails have been processed there is an outstanding loan balance, the loan status is then updated to `NotFullyFunded`. When `NotFullyFunded`, you can submit new funding information.

Skipped rails

If the loan is fully funded prior to the last rail, any rails that weren't attempted are labeled **Skipped**.

Status: Returned funds

In most cases the funding goes smoothly and the funds are fully received by the borrower. However, funding requests can fail immediately or succeed, and then be returned later. If a rail fails, is rejected, or returned, Arix reports a detailed error message as part of the railUpdate webhook and in the `LoanDetail` API.

Loan funding follow-up

Arix records all the funding attempts and continuously updates the status as each rail is processed, funded, or failed. If the loan wasn't fully funded and is in the status, `NotFullyFunded`, you can submit new funding information an unlimited number of times. This is also the process for required multiple disbursements over a period of time for a single loan.

You should use the RailUpdated webhook to follow the funding process.

> Any loan update made via a **/PUT** call is ignored when the loan is InFunding.

Arix generates **webhooks** which keep you up to date on the loan status.

Based on how the funding progresses, the loan can be in one of these statuses:

- `InFunding` leading to `Funded`
- `NotFullyFunded`
- `Returned`

Pull funding from merchant

The **PullFunding API** enables Marketplace Lenders (MPLs) who leverage Arix to manage Merchant Returns. This functionality provides a method for MPLs to retrieve funds from merchant accounts back into the bank, using Nacha debits. This is especially useful when a merchant initiates a return or refund, as it reduces the need for manual reconciliation or additional requests, by reusing previously funded loan.

> It is best practice to wait three days before sending out funds after a successful pull.
>
> According to Nacha standards, a pull can be returned to the merchant. Most returns tend to occur within the first three days. To reduce the likelihood of overfunding, it is advisable to wait before sending out the next `FundingInfo` request.

Next steps

- **Loan seasoning, sale or retention**
- **Servicing data submission and reconciliation**

5.3. Loan seasoning, sale or retention

Loan seasoning

After **funding**, every loan enters a seasoning period that is managed in **Selling**. Cross River provides a range of warehousing facilities, with customized pricing to fit your needs.

Loan sale or retention

At **loan origination** you decide either to purchase the loan back at the end of the seasoning period or have Cross River retain the loan on its books.

The following options are available:

- **Purchase:** The MPL repurchases the loan from Cross River. This process is managed by you via the Lending Selling solution, which offers a user-friendly interface and API integration for efficient loan buyback transactions and investor management.
- **Retention:** Alternatively, Cross River retains the loan, adding it to its portfolio.

Next step

- **Servicing data submission and reconciliation**

5.4. Loan data exchange

MPL submits servicing data on a daily basis. This is required for loans held by Cross River and optionally for those sold back. The data includes principal/interest breakdown, next due date, and delinquency info. The data is submitted via SFTP.

To access the SFTP:

1. **Set up SFTP with Cross River**.
2. Access your SFTP folder using the credentials and information you receive from your Relationship Manager.

> To transfer the files securely, you must use a client that supports SSH with the following public-key algorithms:
>
> - RSA
> - sha2 521
> - sha2 384
> - sha 256

These loan reports are available:

- **Servicing**
- **Transaction**

5.5. New in lending

Selling update - auto purchase and SOFR

Release Date: 2025-07-03

Auto-purchase: MPLs can choose to have the HFS purchase process run automatically without having to log in to Selling UI or call the API. This can be configured to run at a time between 10 AM and 6 PM Eastern.

SOFR API: For partners that use Cross River Warehousing facilities, Selling provides a daily SOFR rate from the NY Federal Reserve. There are currently two available options: The daily SOFR rate as well as a daily 30-day average called SOFRAI.

Graham Agee

Sr. Product Manager, Lending

Webhook update – ACH success message enhancement

Release Date: 2025-05-15

We've enhanced the **RailUpdated** webhook for ACH/ ACHSD rails to include the ACH trace number directly within the content.Message field for successful funding events.

What Changed:

- Before: `"Message": "LOAN"`
- After: `"Message": "Description: LOAN Trace#: 021214890645262"`

Why It Matters:

This addition provides better traceability and aligns webhook data with NACHA record details, making it easier for partners to reconcile and audit ACH disbursements.

No Changes to JSON Structure:

The overall structure of the webhook payload remains unchanged, ensuring backward compatibility.

Ari Pollack

Sr. Product Manager, Lending

Improving B2B integration: Cross River's approach to partner data exchange

Introduction

At Cross River, we are focused on making financial technology more efficient for our partners. We understand that B2B data exchange can be complicated, especially in regulated industries like lending and payments. That's why we are working on a **new approach to partner integration**—one that improves **data validation, schema flexibility, and user experience** to make collaboration smoother.

The challenge: Static APIs and data complexity

Most API integrations rely on **fixed JSON schemas**, forcing partners to adapt to rigid data structures. This can lead to inefficiencies, requiring partners to send redundant data, deal with validation errors, or go through multiple iterations before getting the integration right. Regulatory changes like **1071 compliance** add even more complexity, making flexibility important.

Our solution: A more flexible schema management platform

To address these challenges, we are developing a **new B2B integration platform** that allows partners to define, customize, and validate JSON schemas **more dynamically**. We are starting with our **Lending and Credit Card partners**, who face the greatest challenges in adapting to evolving data and compliance requirements. The key improvements include:

- **Customizable Schema Structures** – Partners can use a **custom schema** that still meets our regulatory and business requirements, avoiding unnecessary workarounds.
- **Real-time Validation & Error Handling** – With **JSON Schema-based validation**, partners receive immediate feedback on data quality before submission, reducing delays.
- **Simple UI for Schema Design** – We are building a visual schema editor that makes it easier for partners to adjust data structures with no need for deep technical expertise.

Schema builder

How lending and credit card partners can get involved

We want to build this with input from our partners. As we refine our integration platform, we are inviting fintech companies and marketplace lenders to help shape the design. By participating, partners will gain:

- **Early access** to our schema editor and integration tools
- **A say** in how validation and structure should work for their specific use cases
- **Faster onboarding** for new product offerings and regulatory changes

Looking ahead

We aim to improve how financial data is exchanged between businesses. With **scalable and adaptable integration tools**, we are making it easier for partners to work with us.

If you’re a Lending or Credit Card partner interested in collaborating, we’d love to hear from you. Reach out to our team and be part of the **next phase of B2B integrations**.

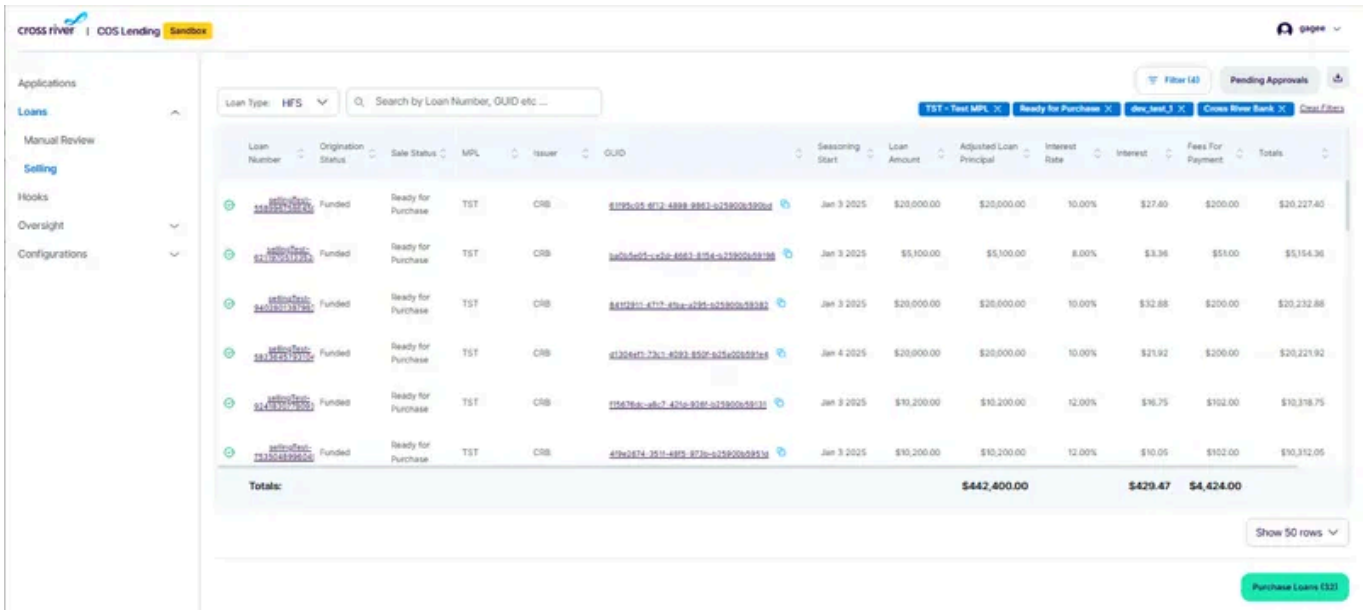
Amichai Lichtenstein

Product, Lending

Introducing COS lending selling

Cross River Bank is excited to announce an enhancement to our Lending process, designed specifically for our valued Marketplace Lending (MPL) partners. The new solution streamlines the process of purchasing loans, delivering efficiency, transparency, and control. After the designated seasoning period, MPLs repurchase originated loans from issuing banks. With our enhanced Lending Selling solution, MPLs can access a user-friendly interface or integrate with our Lending Selling API . Log on in the morning and see loans scheduled to be purchase that day.

Screenshot of log-in showing loans ready for purchase.



Key benefits of the lending selling solution

- **Simplified Loan Management**: MPLs can now log in to our intuitive Lending Selling UI to view and manage loans ready for purchase.
- **Seamless Integration with APIs and Webhooks**: For firms seeking a higher degree of automation, our Lending Selling API enables direct access to the loan buyback process. Additionally, **webhooks** are available to provide notifications and updates, ensuring MPLs stay informed.
- **Pre-Purchase Reports and Post-Purchase Receipts**: Easily access **pre-purchase invoices** and receive **post-purchase receipts** to enable reconciliation and record-keeping.
- **Increased Efficiency**: The new platform minimizes the time and effort required to identify, assess, and complete purchase transactions.
- **Improved Transparency**: With near real-time updates and detailed loan information, MPLs can make informed decisions and maintain control over their purchase process.

Now live for HFS loans, with more to come

We’re proud to announce that the enhanced Lending Selling solution is now **live for Held for Sale (HFS) loans**, enabling MPLs to immediately benefit from this streamlined process.

Looking ahead, we will be expanding the platform to enable management of loans in a warehousing facility , ensuring full coverage to meet your needs.

We’re committed to empowering our MPL partners with tools that make loan buyback processes faster, smarter, and easier. Whether you prefer the hands-on control of our UI or the automation of our API combined with the proactive capabilities of webhooks, Cross River Bank is here to support your success.

For access requests, inquiries, or feedback, please don't hesitate to contact our dedicated support team at **arix.support@crossriver.com**. To access the UI, your user will need *COS Lending Selling* permissions from an allow-listed static IP address.

Graham Agee

Sr. Product Manager, Lending

Introducing manual review UI: A synergistic addition to Arix's lending APIs

We're thrilled to announce a significant milestone for Arix, CRB's loan validation and funding system. For the past five years, Arix has empowered Marketplace Lending (MPL) partners with an API-only integration. Today, we're excited to unveil a complementary addition: Manual Review, a UI designed to enhance the loan review experience and speed up onboarding.

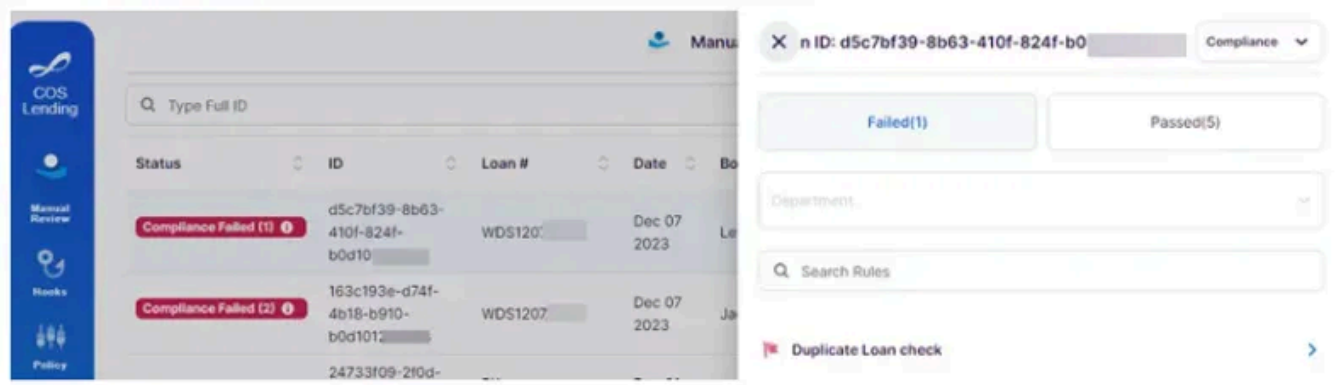
What's new?

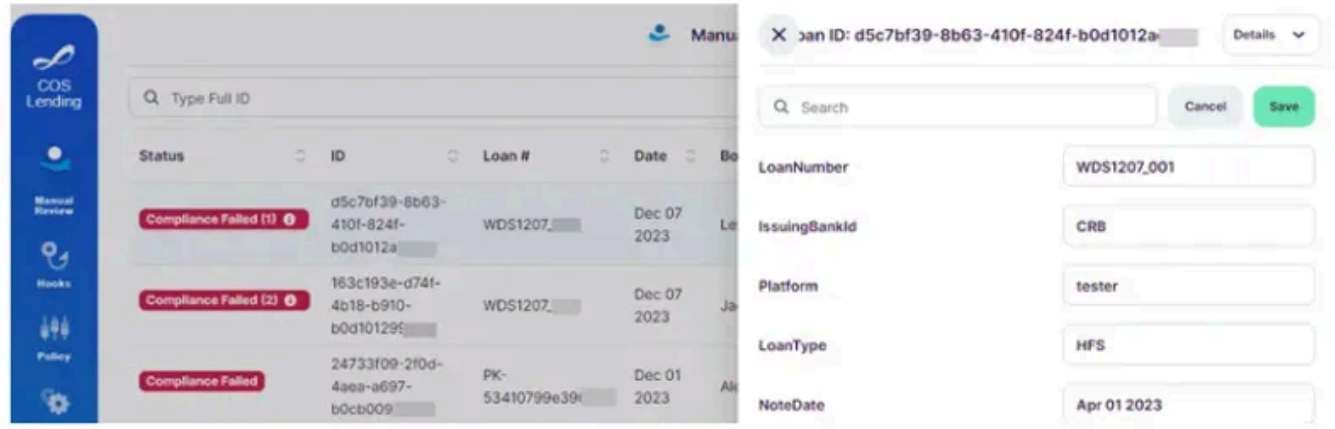
In response to your feedback, we've developed Manual Review, a UI that integrates with our API-driven system. Now, you have the flexibility to choose between API calls or intuitive UI interactions for loan management tasks. The system currently shows all the loans that have not automatically proceeded to funding.

Key features

1. Dashboard: Gain comprehensive visibility into loans that are in the queue.
2. Actionable insights: Easily identify why a loan is in the queue, edit loan details or upload documents directly from the UI. These actions rerun the compliance engine.

3. Seamless transition: As an existing user, enjoy consistency and ease of use because all functionalities available via the UI mirror the capabilities of our Arix API.
4. Enhanced collaboration: Add notes to a specific loan, giving Bank Ops more information needed to pass a failed rule (coming soon)!





As we celebrate this milestone, we're committed to further enhancing the Arix experience. In future releases, expect even more features and refinements aimed at optimizing loan processing efficiency and compliance management.

Experience the fifference today!

For access requests, inquiries, or feedback, please don't hesitate to contact our dedicated support team at **arix.support@crossriver.com**. To access the UI, your user will need **Oversight** and **Loan Edit** permissions from an allow-listed static IP address.

Here's to a future of seamless lending with Arix!

Ari Pollack

Sr. Product Manager, Lending

Streamlining loan funding testing with “BypassRuleChecking”

Building and optimizing a lending platform takes time—and testing should be frictionless. At Cross River, we’re making it easier for our partners to `validate loan funding and loan purchasing flows` **without triggering full compliance checks** in the sandbox environment.

Now, by adding the "`BypassRuleChecking`" key in the **Field5** JSON, you can bypass compliance rule checks during sandbox testing. This lets you focus on funding mechanics and iterate faster, without waiting on full verification cycles.

Want to see it in action? Check out our **Loan attributes and parameters** for details.

As always, we’re here to help. Reach out with any questions.

Amichai Lichtenstein

Product, Lending

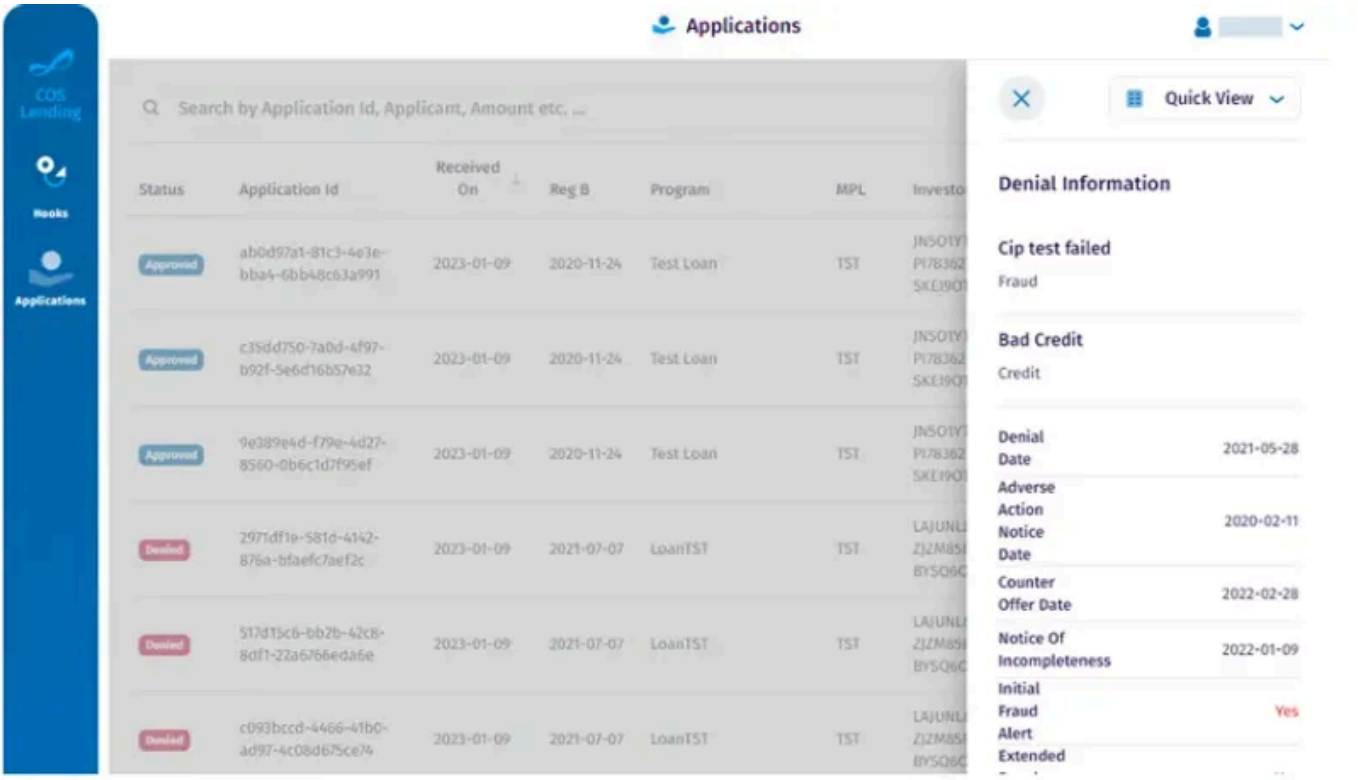
Streamlining fair lending requirements with the Pre-approval UI

We’re excited to announce the launch of our new web-based app for viewing loan application data that’s sent using our Fair Lending API set. This new user interface (UI) lets you view, sort, and filter loan application data, making it more efficient to access the information you need for successful implementation of our APIs.

One of the major benefits the UI gives is the ability to view a large amount of data without the need to pull it via API. You can access the information you need without having to deal

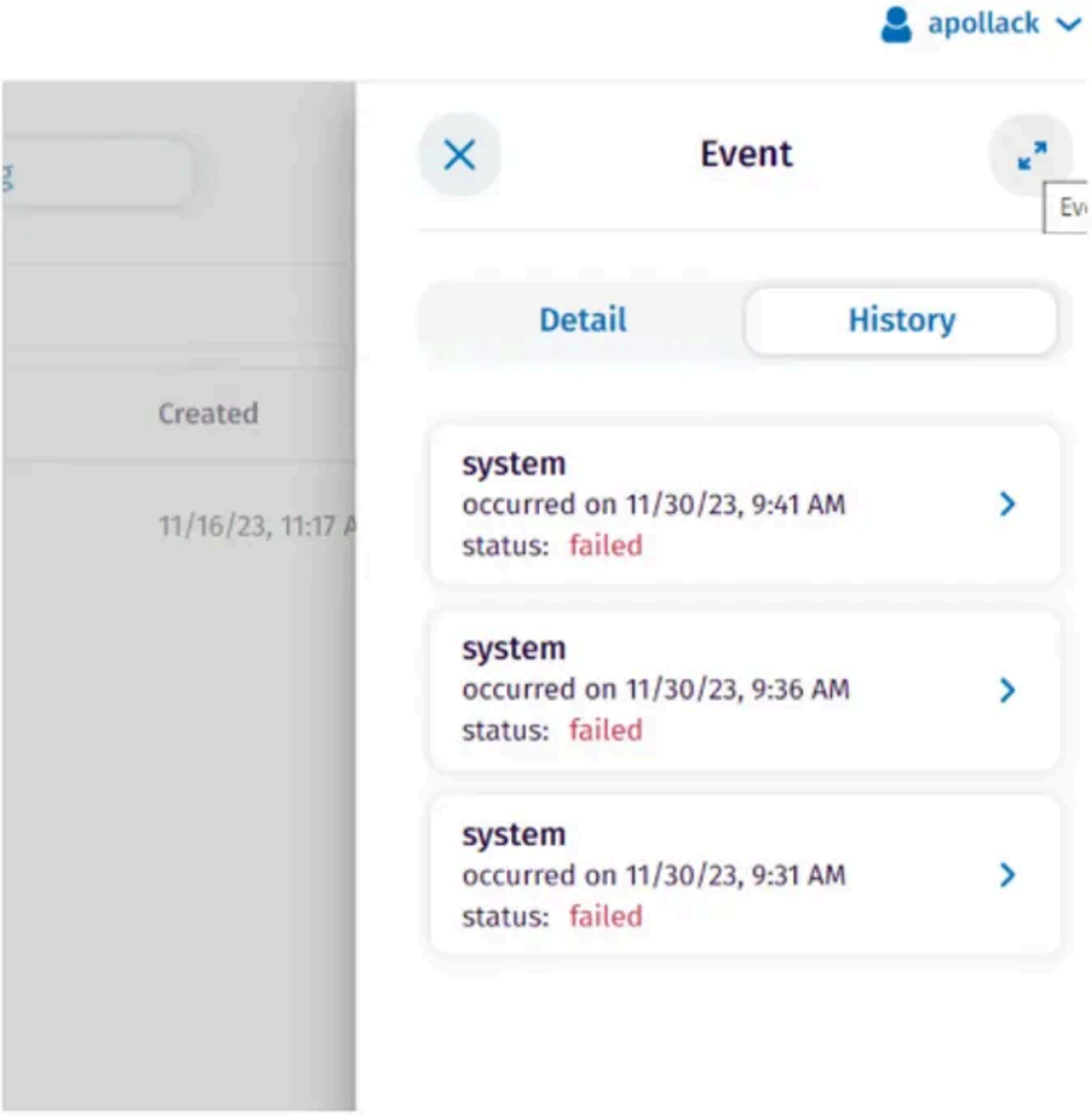
with any technical details or coding. The app is ideal for anyone who needs to review loan application data on a regular basis. It also makes onboarding to our systems easier.

Here's a view of a denied application, including the reasons the loan application was denied.



Another benefit the new app gives is the ability to navigate through information, without being overwhelmed by the amount of data needing to be collected. This is sometimes hard to do in a single large JSON object.

This image shows a drill down into an approved application that's been accepted by the customer.



Overall, our new UI for viewing loan application data is a valuable tool for anyone who needs to access this information. Whether you're a loan officer, Fair Lending specialist, or just someone who needs to view data as part of your job, this new web-based app makes it easy to get the information you need quickly and efficiently.

Ari Pollack

Sr. Product Manager, Lending

New mailing options for checks

Hey API Developers, Engineers, and Product Managers!

Great news from CR for Arix users! We're releasing an update to the check rail, offering more mailing options:

1. FedEx Express Overnight
2. FedEx Priority Overnight
3. FedEx Express 2 Day
4. USPS Overnight
5. USPS Priority "2 day" (delivery confirmation)
6. USPS Priority "2 day" (signature required)
7. USPS First Class - Canadian
8. USPS First Class - International
9. USPS First Class - US
10. USPS Certified Mail

You now have the flexibility to choose the best option for your needs. Plus, for auto loans, checks will expire after 40 days, giving you more time to process payments.

CR values your feedback and is committed to improving your experience with Arix. Head over to check out these fantastic updates and stay tuned for more!

Happy lending.

Amichai Lichtenstein

Product, Lending

Expanding possibilities: Arix welcomes wires and card payment rails to the family!

Arix is thrilled to introduce two new rails to our **collection**, bringing our total to eight!

1. **Wires.** Effortlessly wire funds directly through Arix. Pricing will remain the same if you are using wires through COS today.

2. **Card payments**. The Card Payment Rail allows you to send funds directly to credit cards. This can be used as a failover for RPPS rail for covering a wider network, or a stand-alone rail.

These rails can be utilized through the Arix APIs, just like one would call the other payment rails.

The case against the two-step solution

The wide range of payment rails wasn't always available directly in Arix. Workaround solutions were found by using the XPay rail for internal transfers and then sending funds from your account using another payment method.

It's time to ditch the two-step solution.

Here's why:

1. Efficiency Matters
   The two-step process introduces unnecessary complexity and increases the chances of errors. Direct wiring and card payment rails from Arix simplifies transactions, reducing manual reconciliations and potential failures.
2. Capture Every Loan Funding Event
   With Arix APIs, loan funding events can be seamlessly captured in one step, and in one system. No need to juggle between different systems, Arix ensures a comprehensive record of all your loan disbursements.

Ready to Transform Your Transactions?

Upgrade your loan processing experience with ArixWires and Card Payment Rails. Explore the possibilities of these new additions, now totaling eight rails in our comprehensive system.

Learn more

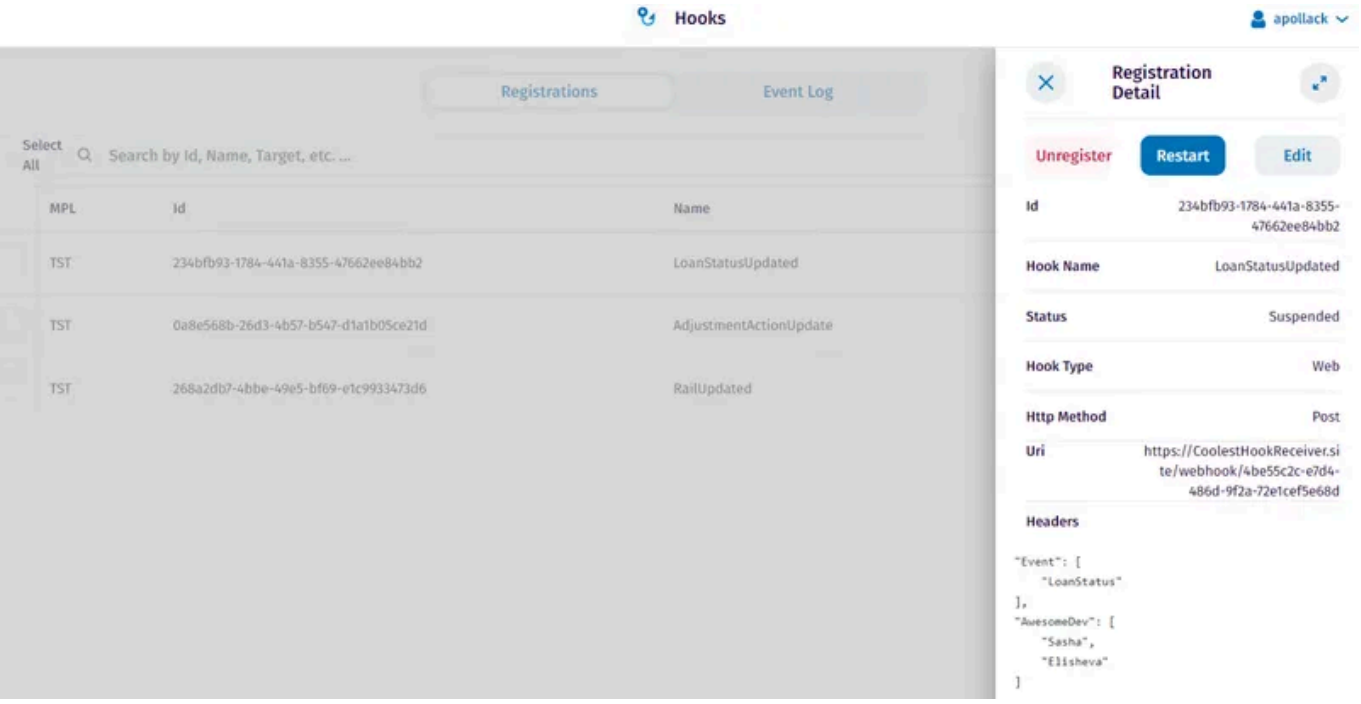
Curious about how to sign up or want to explore these new features further? Reach out to your dedicated relationship manager today. They're here to guide you through the seamless integration of these rails into your loan operations.

Ari Pollack

Sr. Product Manager, Lending

Unveiling the new Hooks UI: Elevating your connectivity experience

We're thrilled to introduce a transformative update that takes our client-facing UI to new heights! Say hello to The New Hooks UI, where functionality meets self-service, and your control over notifications is taken to the next level.

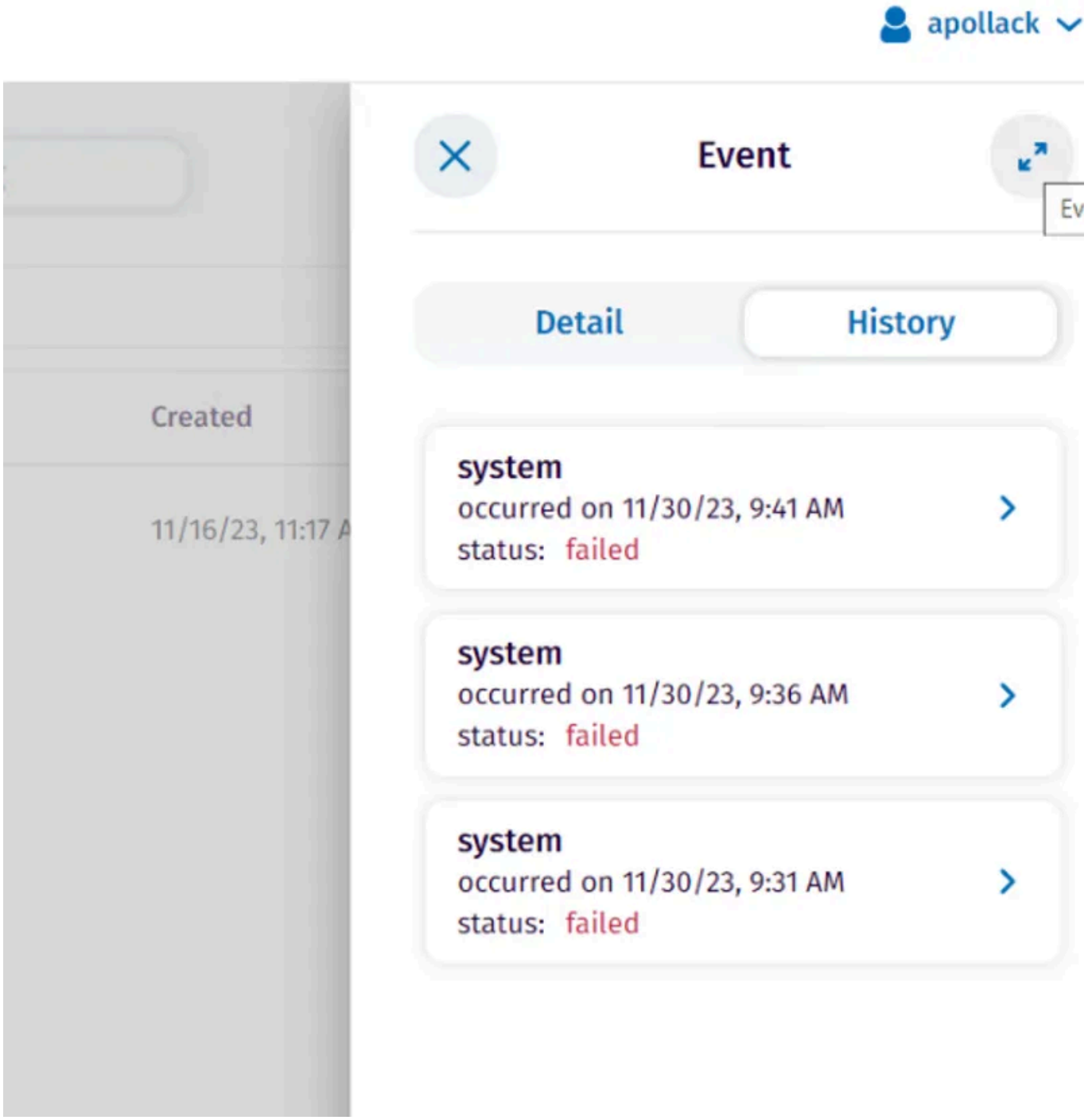


Registrations can be restarted, so all failed events are resent.

Empowering features

1. Interactive Connectivity: You can now create registrations for events, ensuring a tailored experience.
2. Troubleshoot with Ease: Encounter an issue? No problem. Troubleshoot historical data of events, dive into requests sent, and inspect responses from your system – all within a single, user-friendly interface.

3. Edit Registrations and Resend Events: Fine-tune your interactions effortlessly. Edit registrations, restart processes, and resend events on your terms.



Empower your team, save your resources

The New Hooks UI isn't just a UI update; it's a strategic move to empower your development team and save valuable resources. Faster troubleshooting means more efficient workflows, translating directly to cost savings for your company. The ability to resend events means that no events are lost, and even if the network is down, your system can be updated at a later time.

Ready to experience the future of connectivity? Dive into The New Hooks UI today!

Your Connectivity, Your Control.

Ari Pollack

Sr. Product Manager, Lending

Streamlining fair lending

My name is Ari Pollack and I'm a Product Manager on the lending team. I'm excited to announce the release of our newest set of APIs, Cos.Lending PreApproval. At this first stage, you can send application data to Cross River for all applications (originated, approved-not-accepted, denied, withdrawn, rejected, incomplete, and if applicable, in progress).

Why is this necessary?

Regulatory agencies across the industry recognize how unique our business model is, and have been scrutinizing FinTech lending activities as it relates to Fair Lending. As a partner, expect an annual examination as regards fair lending, in addition to regular examinations. The bank's prudential regulator, the FDIC, expects us to evidence collaborative data management practices between Cross River and its partners.

Validation of data on submission

Until recently, only approved and ready to fund loans were sent via API to the system. All other application information was sent daily, as files via sftp. Partners often received feedback from regulators, months and even years after the data was sent in which led to burdensome research to fix old data. With the new Application API, we validate each field on submission, and return instant feedback in the response of the API call. Required fields are enforced so that an application can be recorded. For example, the Approve API in the image below where an application is being sent in and marked as **Approved**.

POST https://lendingsandbox.crbcos.com/preapproval/v1/applications/approve

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
287            "citizenship": "US",
288            "residencyType": "IllegalAlien",
289            "phone": "7802088592",
290            "email": "Budcrbtest.com",
291            "addresses": [
292                {
293                    "street": "935 Strosin River",
294                    "city": "Kassulkeberg",
295                    "state": "SC",
296                    "country": "US",
297                    "postalCode": "27313a"
298                }
```

ody Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1  {
2      "Version": "1.0.0.0",
3      "IsSuccessful": false,
4      "StatusCode": 400,
5      "Errors": [
6          {
7              "Code": 1000,
8              "Message": "Validation error(s) on field Borrowers[0].Email: [\"Budcrbtest.com is not a valid email\"]"
9          },
10         {
11             "Code": 1000,
12             "Message": "Validation error(s) on field Borrowers[0].Addresses[0].PostalCode: [\"27313a is not a valid PostalCode for country US\"]"
13         }
14     ]
15 }
```

In the API response you can see immediate feedback. The borrower had an illegal email address and the postal code was also not valid. Now, the submitter will know what needs to be fixed much earlier on in the process.

Denial API

For Fair Lending purposes, we are not only required to collect information about approved loans, but also for denied, rejected, or undecided applications. Below is an example of an application submission to the API with a decision of Denied. As you can see in the image, there is a list of denial reasons in the body of the API call along with the application itself.

POST https://lendingsandbox.crbcos.com/preapproval/v1/applications/deny

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON


```
447        }
448    },
449    "applicationDeniedRequest": {
450        "denialReasons": [
451            {
452                "declineReason": "The borrower is a known fraudster",
453                "policy": "Fraud"
454            },
455            {
456                "declineReason": "The credit score is outside of the credit policy parameters",
457                "policy": "Credit"
458            }
459        ],
460        "denialDate": "2021-05-28",
461        "adverseActionNoticeDate": "2020-02-11",
462        "counterofferDate": "2022-02-28",
463        "noticeOfIncompleteness": "2022-01-09",
464        "initialFraudAlert": true,
465        "extendedFraudAlert": true,
466        "addressDiscrepancyFlag": true,
467        "creditFreeze": true,
```


Long term benefit

Ultimately, this API strengthens data governance and regulatory exam management processes.

While it will take some level of effort to integrate, the benefits of the API process include:

- Eliminates or minimizes data errors
- Enables an automated, real-time process to correct data errors and avoid burdensome research/ correction of errors after the fact
- Facilitates proactive identification and management of compliance and fair lending risks
- Allows more timely responses and adequate preparation for regulatory examinations
- Ensures adherence to record retention and technical compliance requirements

The FDIC expects to receive complete and accurate data in a timely manner during the regulatory examination process. The API process will go a long way in meeting those

expectations and minimizing potential scrutiny by the examiners, which helps everyone involved.

Ari Pollack

Sr. Product Manager, Lending

Check's in the mail

Don’t start eulogizing the paper check just yet. Our “Check in the Mail” keeps gaining popularity. This release allows for scalability, security, and a better user experience for our partners.

At the end of 2020 we launched our Check in the mail rail. MPLs use checks to give their customers the option of receiving funds or paying off their existing loans via a paper check. Very old school, yes, but our MPL customers jumped at the option.

Our vendor, Checkbook, did a great job of printing and sending out the checks. Because of our architecture, they also managed the check clearing. As the volume increased though, relaying details of cashing, revoking, and expiring checks became a non-scalable procedure.

Positive pay and COS clearing

To address that issue, we needed to automate the clearing of the checks in-house. COS, our banking core, added a Positive Pay feature, and once the system integrated that feature, check events could be automatically handled.

Cashing a check

When a check is created, the system adds it to Positive Pay in COS and it has an expiration date. The check is then printed and mailed out by Checkbook. When a customer tries to cash the check, the fed notifies CR, and COS reviews the check. The check only gets cashed if all 3 conditions are met:

1. The check is on the Positive Pay list and matches the account information.
2. The check hasn't expired.
3. The check hasn't been revoked.

The system can revoke a check if MPL Operations or the MPL wants to stop a check from being cashed. And, if any of the 3 conditions are not met, COS will bounce the check when someone tries to cash it.

Return events

Any event that stops a check from being able to be cashed, such as an expired event or a revoke, will create a return event for that check. After a return, you'll be able to request the funds again by check, or by another payment method.

Should I state the obvious? OK, I will.

Since the fed will now send the cash request to CR, cashed events will now be in real time. No tech changes required by our MPL customers. Relax and enjoy the improved experience.

You may notice these features that we added:

- Check number is now printed in the message field for easier tracking
- Check image is now available to our MPL Ops once it is presented to be cashed

Any failure to create a check will result in a return. This will allow the rest of the loan amount to be disbursed using a different rail.

We are working with Checkbook on other possible mailing options (at present, only USPS).

If you are an MPL that's keen on disbursing via digital rails, of course we support those as well as payment rails. If a business opportunity arises that requires more traditional payment methods, we've got you covered. We're not burying the paper check just yet.

Ari Pollack

Sr. Product Manager, Lending

6. Digital banking

Cross River Digital Banking provides businesses with the convenience and flexibility to apply for and manage their bank accounts entirely online. Our digital platform provides a seamless and efficient banking experience, empowering them to take control of their finances anytime, anywhere.

Important concepts and actions:

- **Log in**
- **Send ACH or Wire**
- **Create payment template**
- **24/7 internal settlement**
- **Transfer funds**
- **Manage recipients**
- **High yield commercial savings**
- **Reports**

6.1. Log in

You can set up your company account(s) within the digital banking platform to conform to your company's needs, requirements, and policies, allowing for further customization and control.

Configuring your account - Policies, Security and Control

Cross River's digital banking platform offers a number of security controls to ensure your accounts with Cross River will meet your internal account management policies.

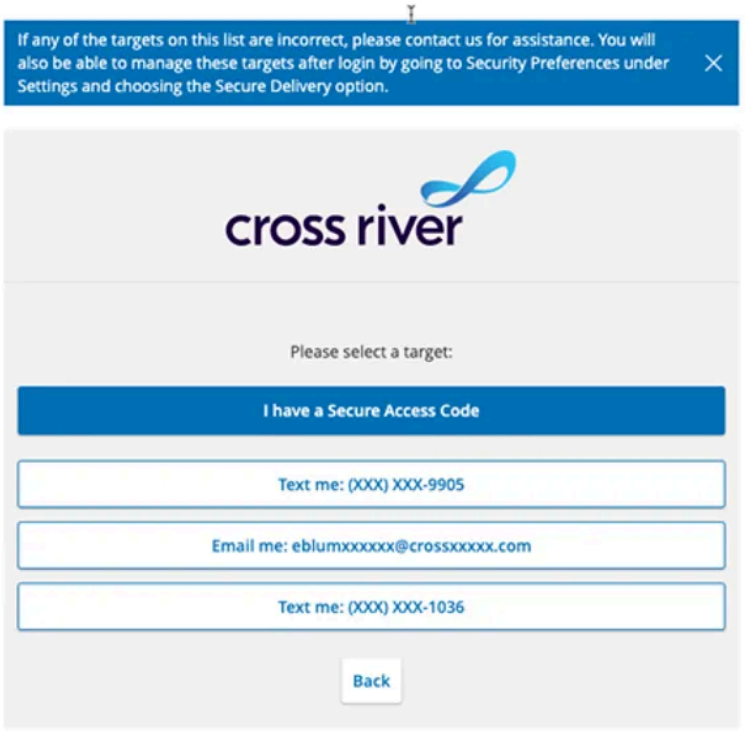
This includes requirements for dual authorization and dual action.

- **Dual Authorization -** This feature requires the user to provide a Secure Access Code in order to complete an action such as logging in to an account, submitting a ACH/Wire transaction, or approving a drafted ACH/Wire transaction. The Secure Access Code is sent via email, call, or text message, as selected by the user.
- **Dual Action -** This feature allows for users to require approval flows for all transactions submitted from Cross River's digital banking platform. At least two users must be involved in each transaction submission (one will draft the transaction, the other will approve the transaction).

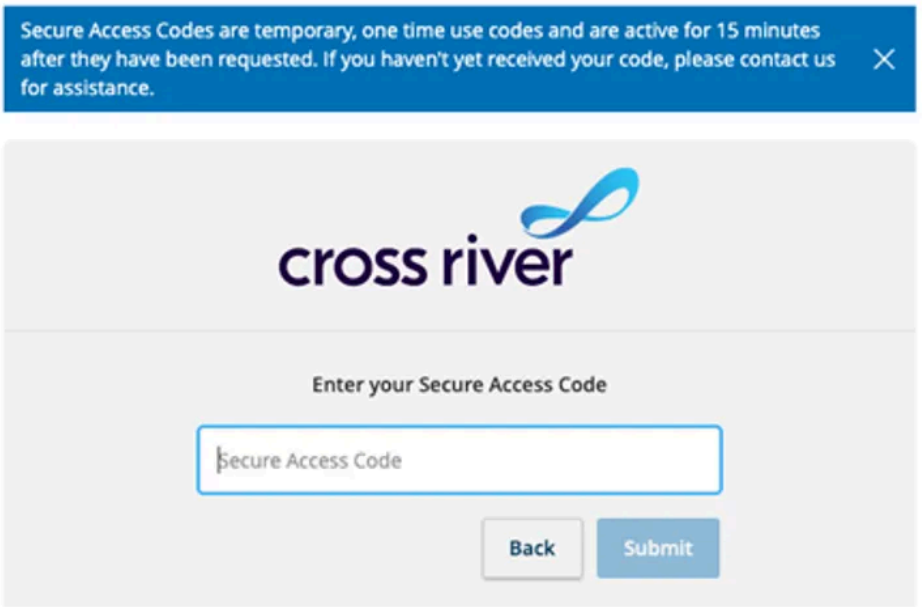
The security controls are configured by the Cross River Ops team. Speak to your Relationship Manager to ensure you have the set up that is right for your organization.

Logging in to the platform

1. Go to **the digital banking platform login page**.
2. Enter the username and password you received in the email from your administrator. You're prompted to select how you want to receive the Secure Access Code.



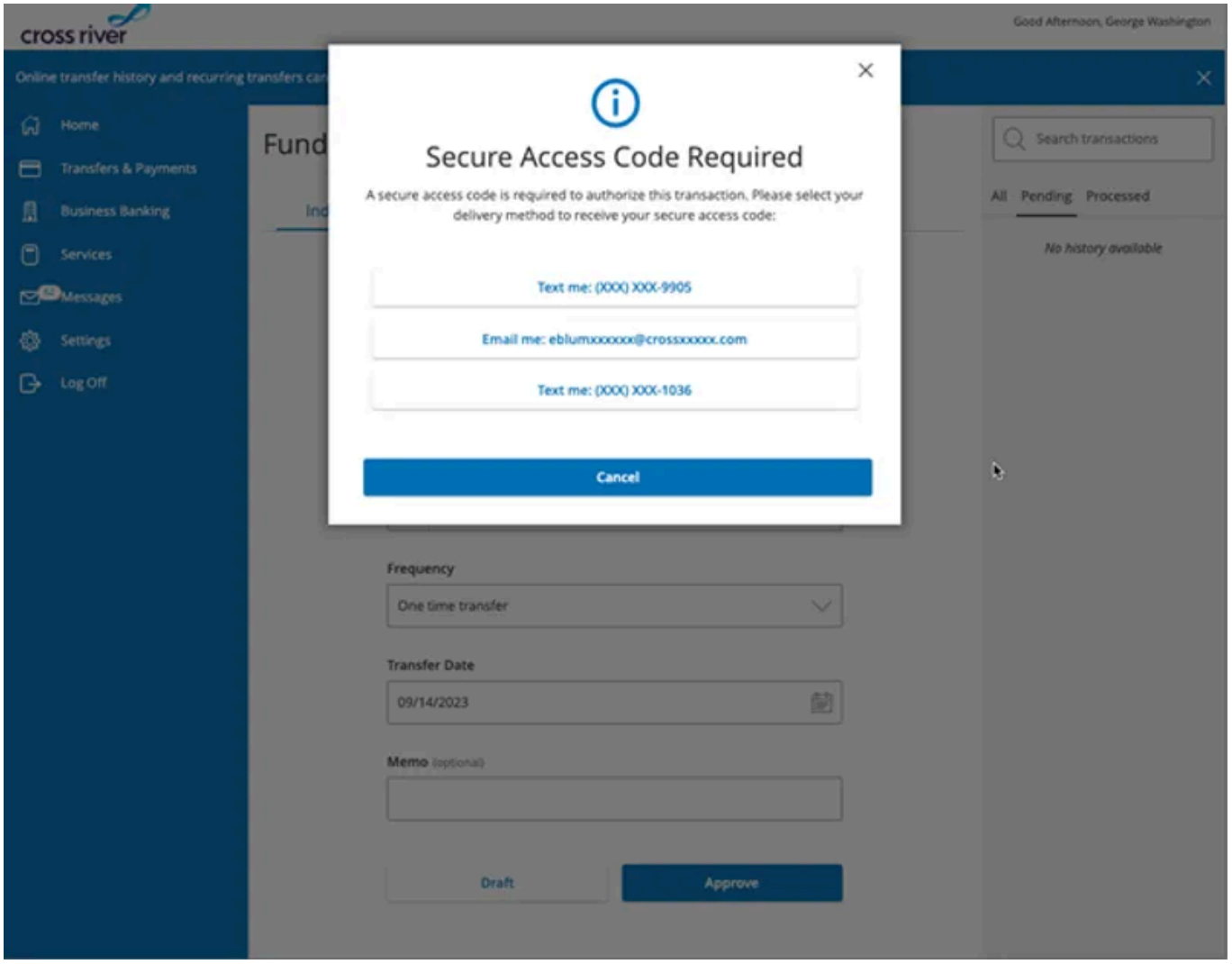
3. Enter the Secure Access Code you received to access the platform.



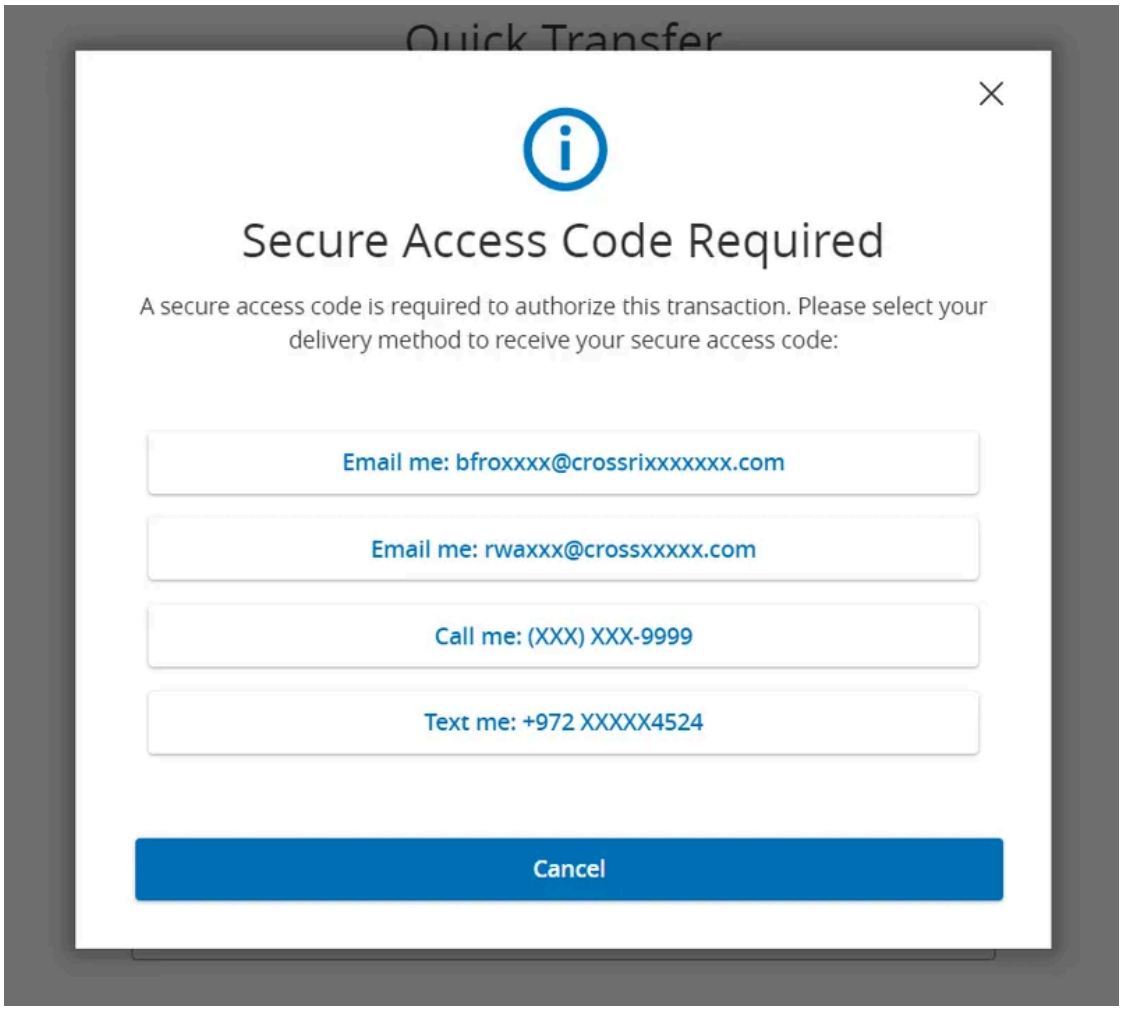
6.1.1. 2 Factor Authentication (2FA)

(2FA)Our system authenticates clients and authorizes them to make transactions.

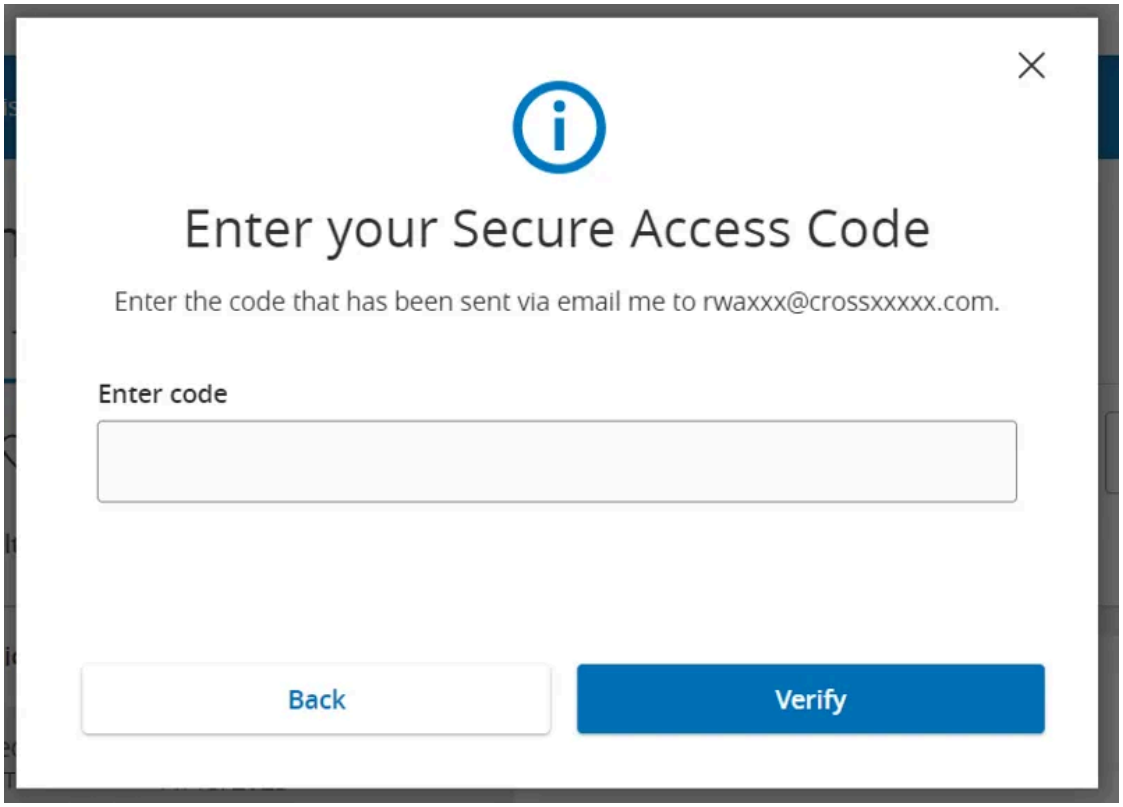
We do this by sending you a **Secure Access Code**, which is triggered by specific transaction requests.



You have the option to receive this code by email, text, or a phone call. We configure the options according to your preferences.



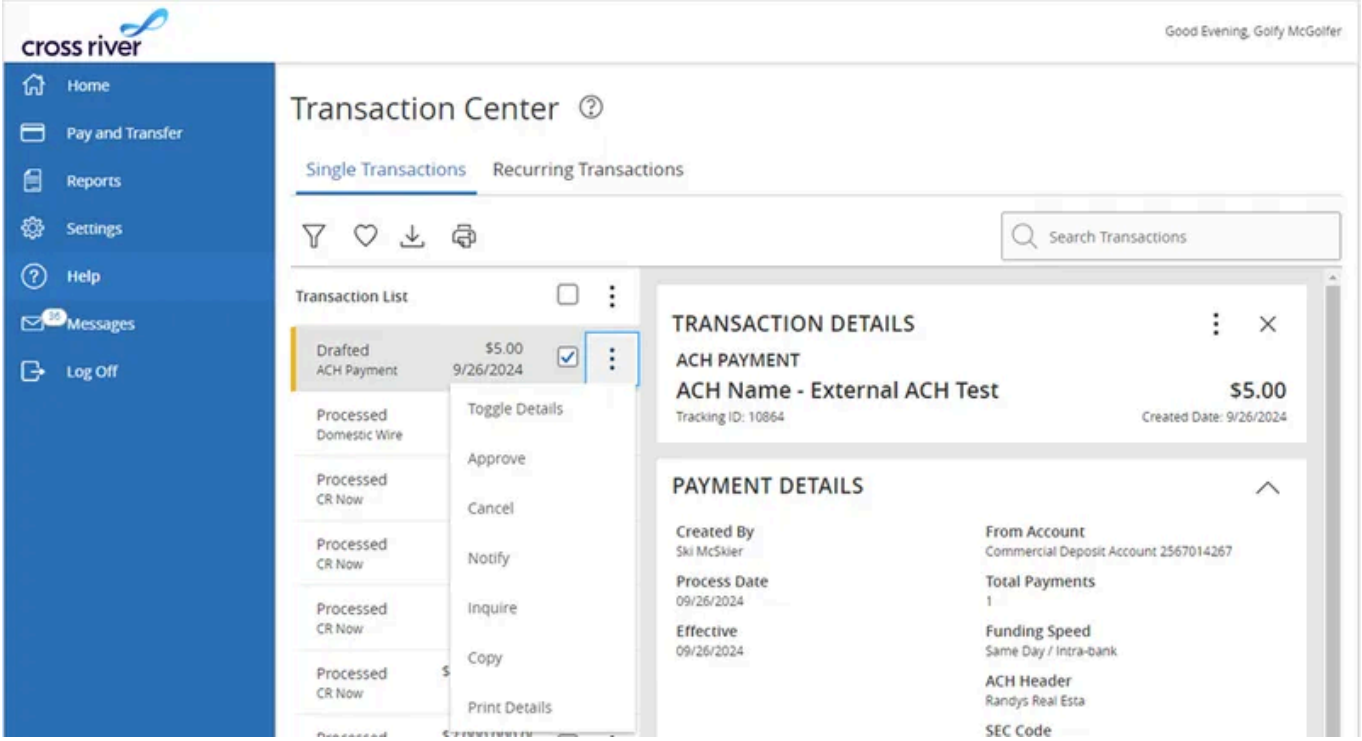
Input your **Secure Access Code** and click **Verify**



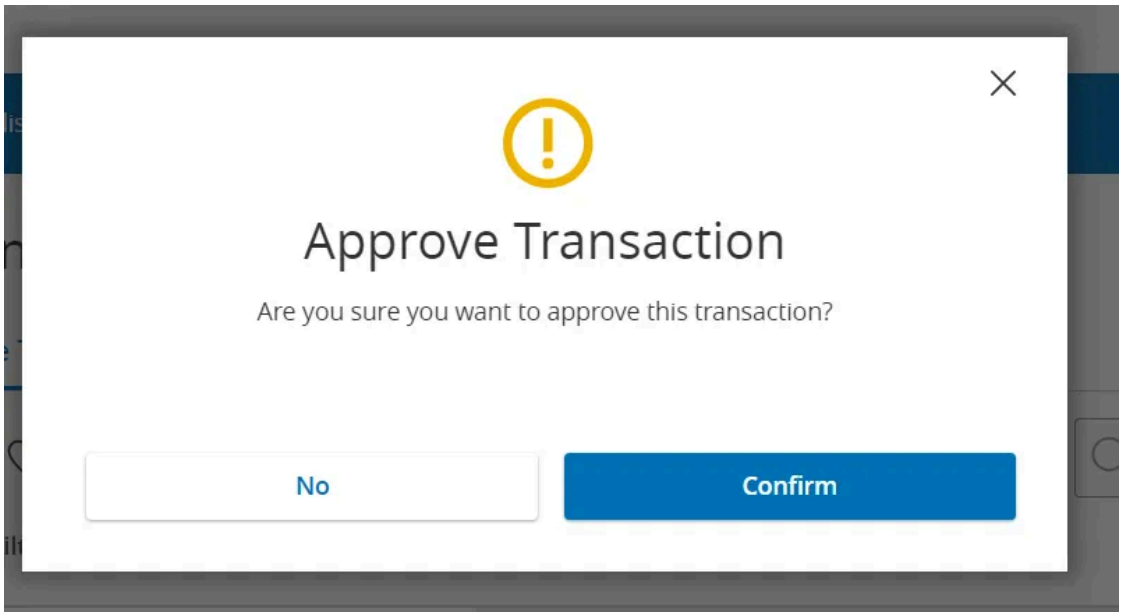
6.1.2. Dual Action

Dual Action, commonly referred to as Dual Control (or Maker/Checker), is offered to all Cross River Digital Banking customers who use the online banking platform. This feature is a strong fraud prevention mechanism that requires two separate people to initiate and authorize every transaction originated from your account. The **Dual Action** feature is configured when we set up your account. Your account admin can decide who can draft transactions and who can approve transactions.

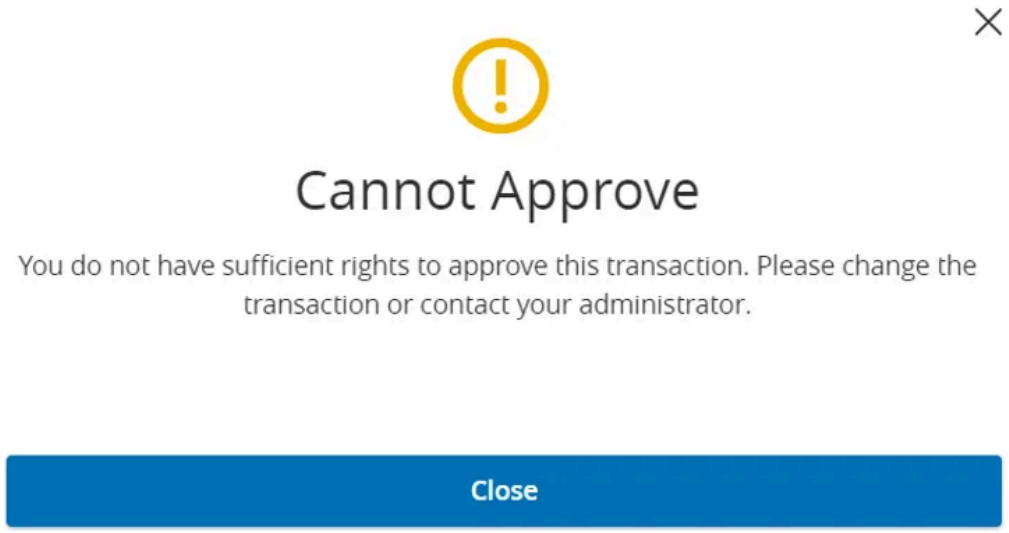
As a “Drafter” you will not have permission to approve your transactions, but can only click **Draft** on the ACH or Wire payment screens. As an “Approver”, you will be able to approve drafted transactions by navigating to the **Transaction Center**, clicking on the 3-dot menu, and selecting **Approve**.



The **Approve Transaction** window displays.

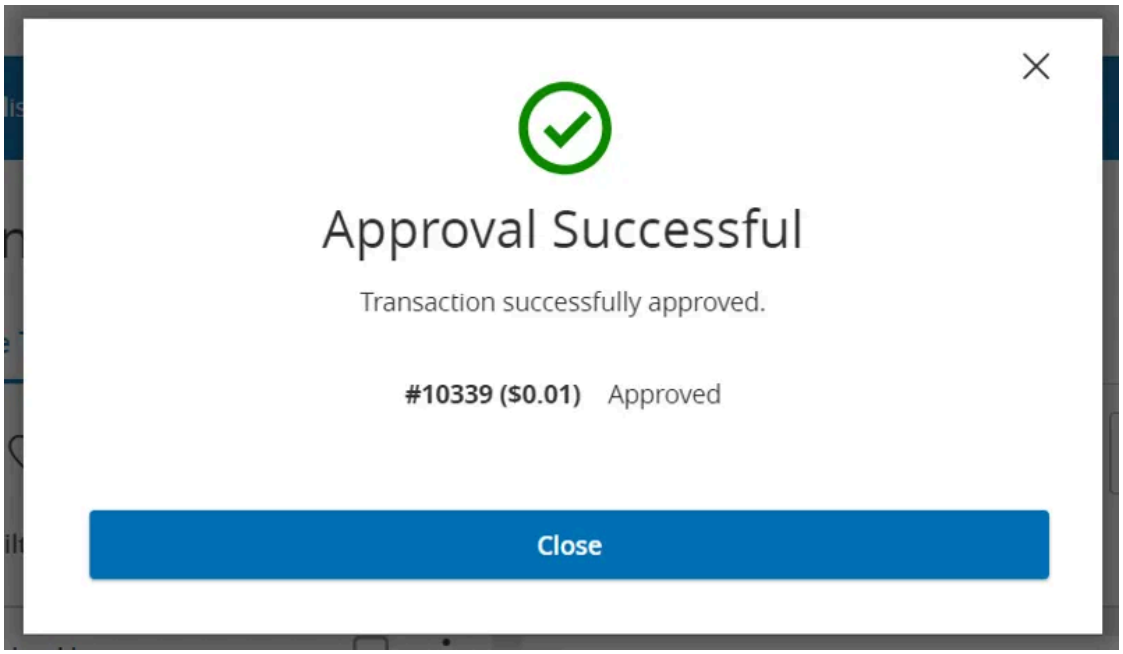


If you are not one of the configured users for approval, then this screen displays.



If you are a configured to approve transactions when you click **Confirm**, this triggers a 2FA.

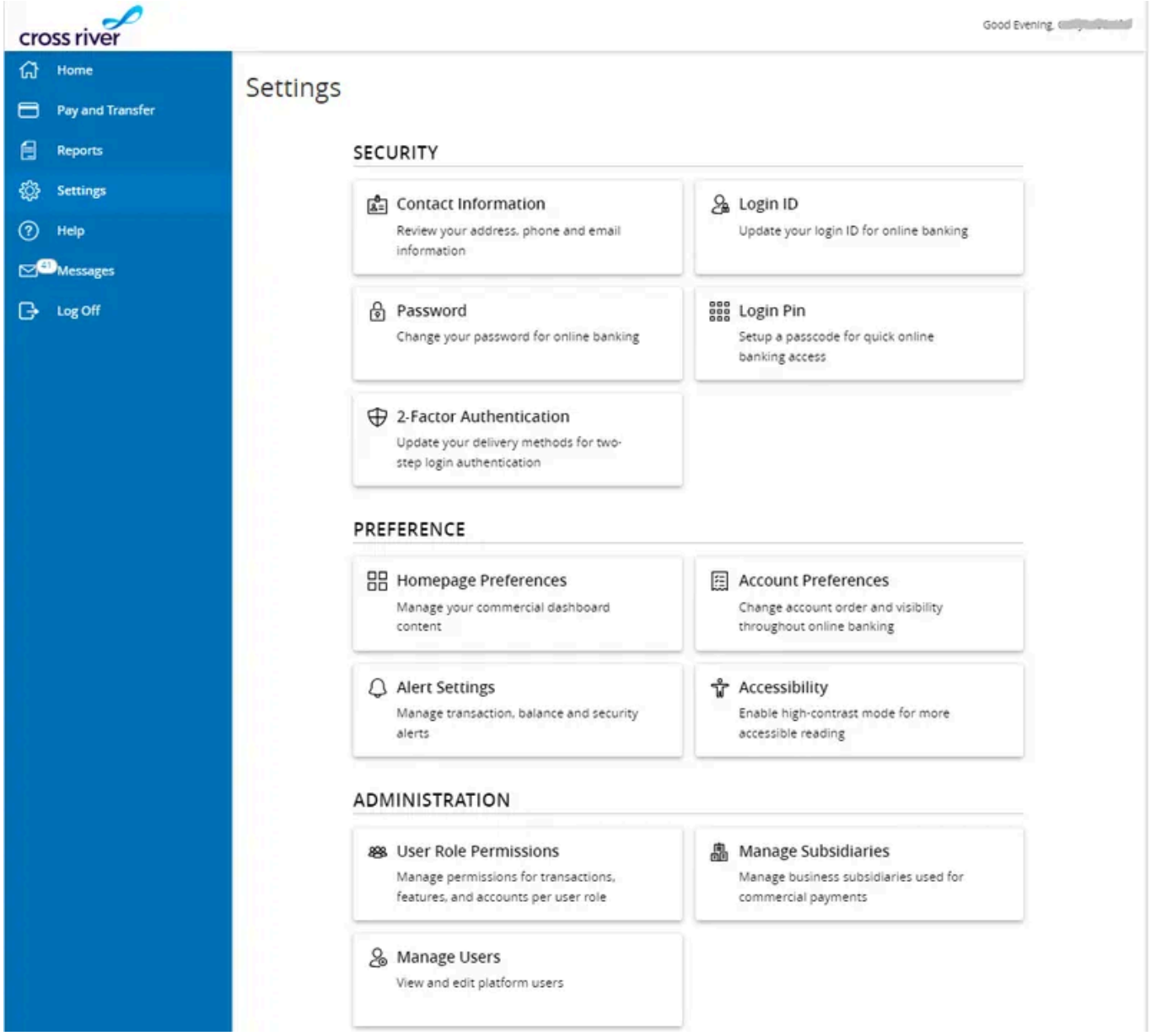
Enter the code (shared with you by email/text/call) and then this screen will display.


Approval Successful
Transaction successfully approved.
#10339 ($0.01) Approved
Close

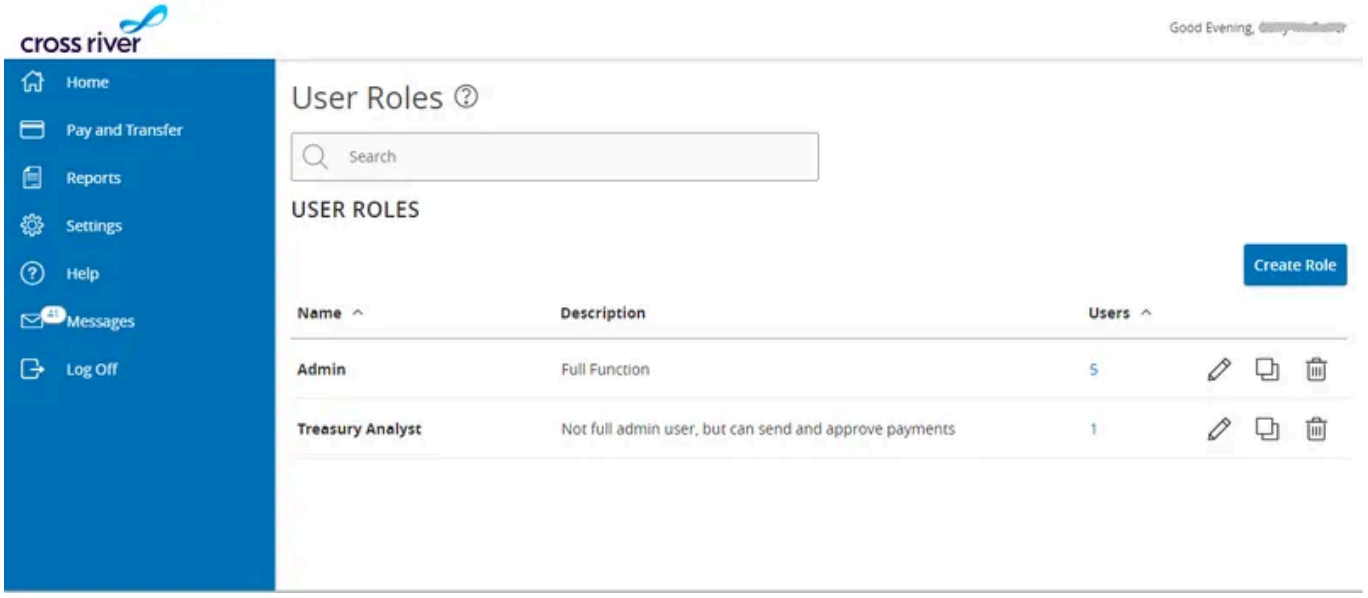
6.1.3. User roles

User Roles allow your company to set different levels of account permission, based on roles determined by the admin user. For example, you may want people in certain roles to only view specific accounts your company has at Cross River. This is all easily configurable in the **User Roles Permissions** tile. You must select a **User Role** for each **User** that you add to your Cross River digital banking account.

In the **Settings** tab, you can click on the **User Role Permissions** tile to access the **User Roles** policy menu.

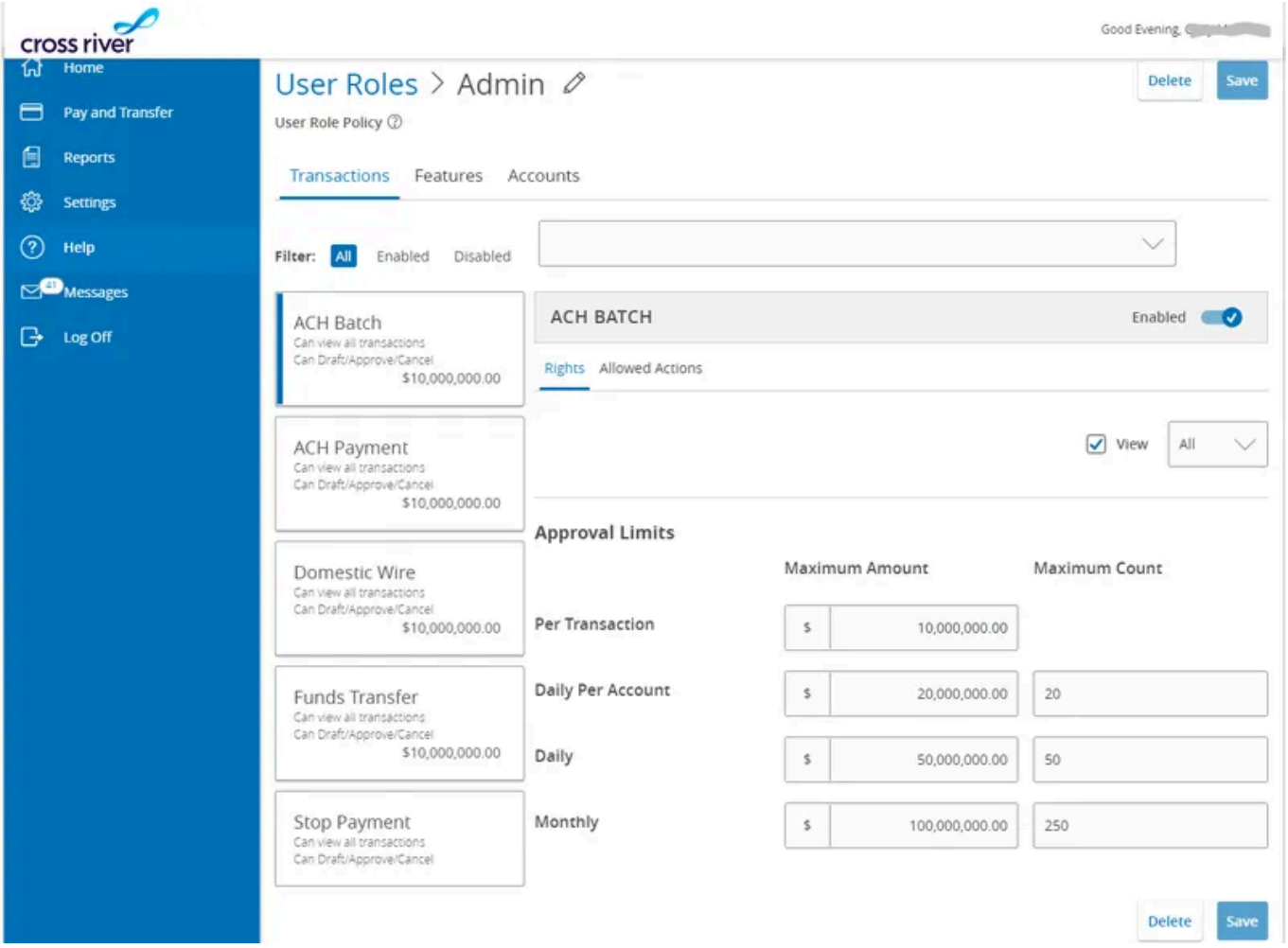


Here you can add new **User Roles** or edit permissions of existing roles. Click the pencil icon to edit an existing role, or click **Create Role** to add a new one.



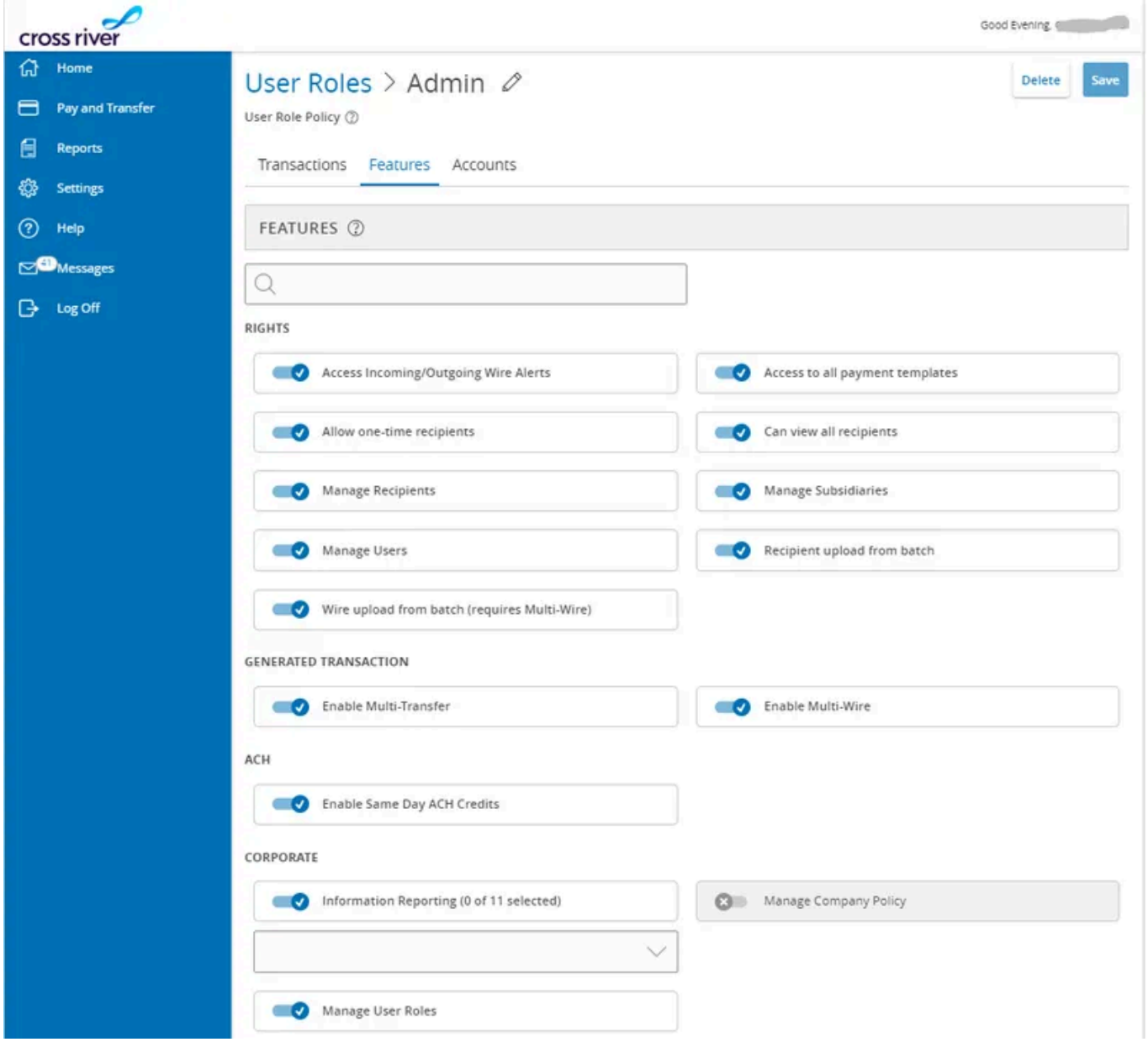
Each user role has associated policies. Any changes or additions made to a **User Role** can only be within the limits pre-assigned and agreed upon during account setup.

The screen displays with 5 tiles. **ACH Batch, ACH Payment, Domestic Wire, Funds Transfer** and **Stop Payment**.

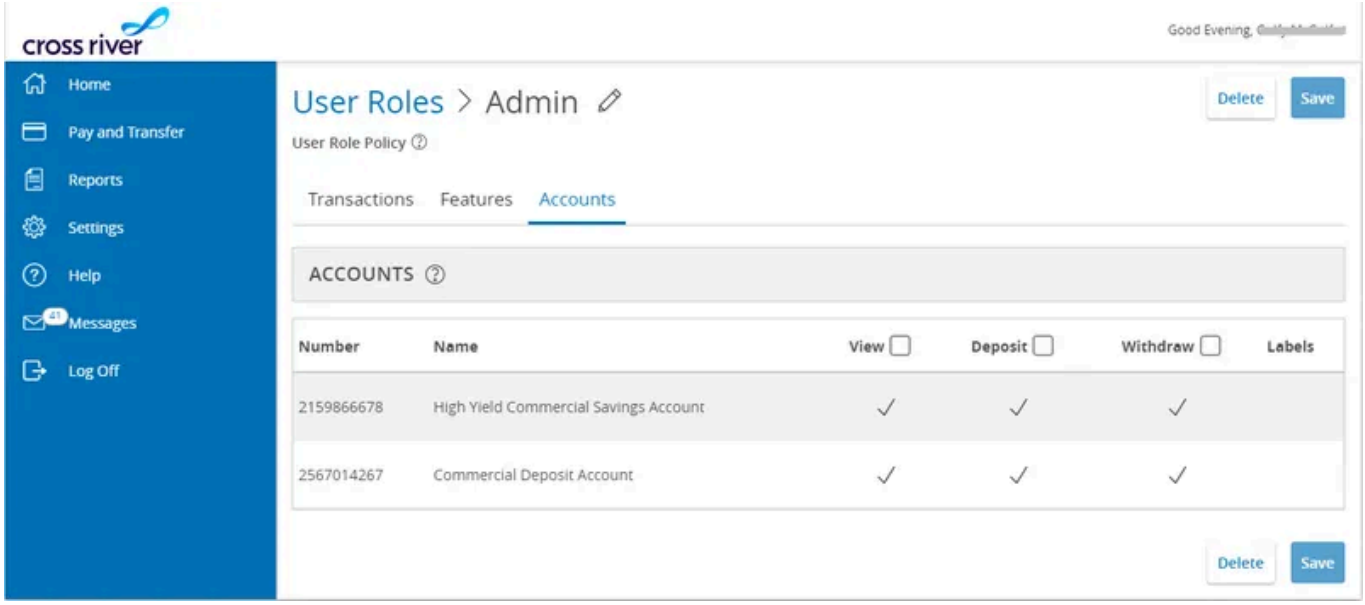


In the **Transactions** tab, there are details of the limits (**Rights**) and **Allowed Actions** that are currently set for that user, relating to transactions. You can edit these **Rights** and **Allowed Actions** for each role in this menu.

In the **Features** tab, you can enable and disable specific features associated with the role.



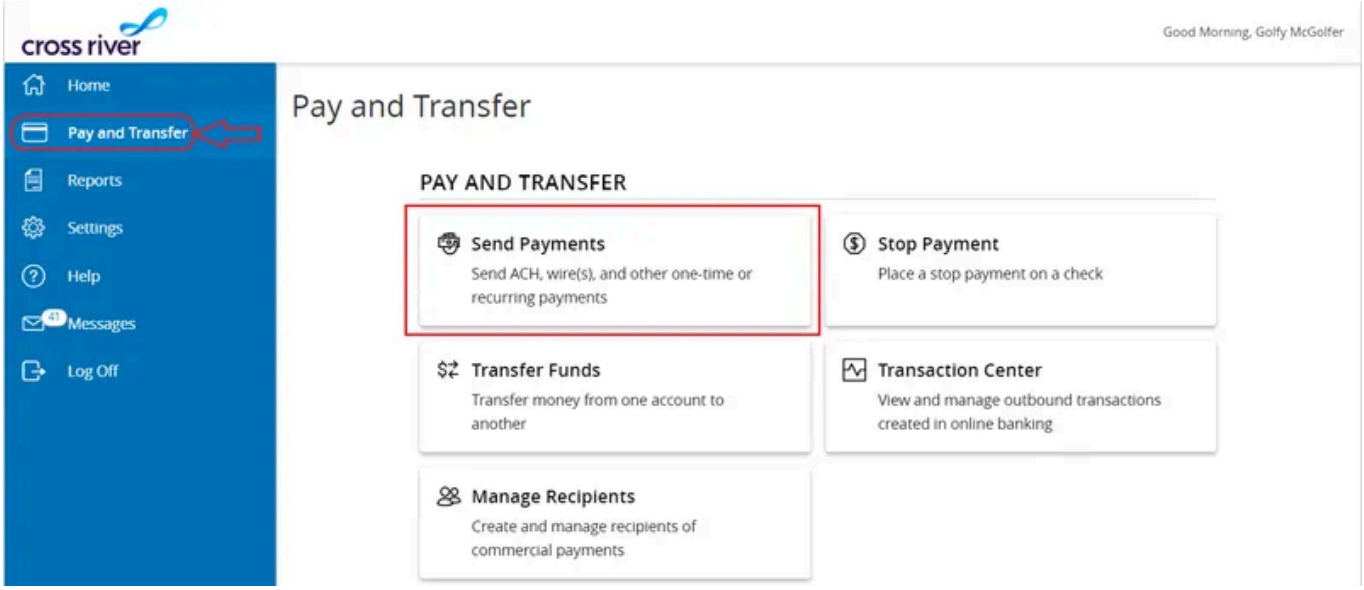
In the **Accounts** tab, you can edit visibility and deposit/withdrawal permission for each account, associated with the role you are editing.



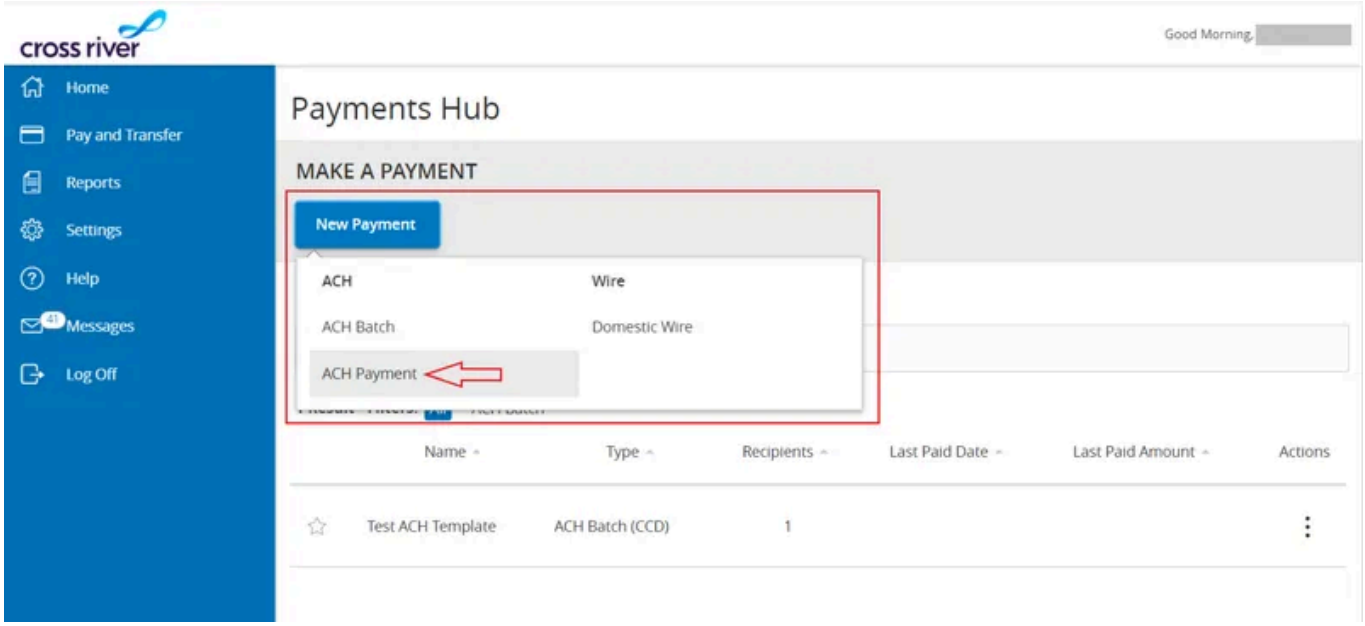
After making changes to the details of the **User Role**, click **Save** to update the edits to that specific role.

6.2. Send ACH or Wire

From your Cross River account, you have the ability to send ACH and Wire payments to external accounts. Currrently, there are three payment types to choose from: **ACH Batch**, **ACH Payment**, and **Domestic Wire**. To view these options you will need to navigate to the **Pay and Transfer** tab, and click on the **Send Payments** tile.



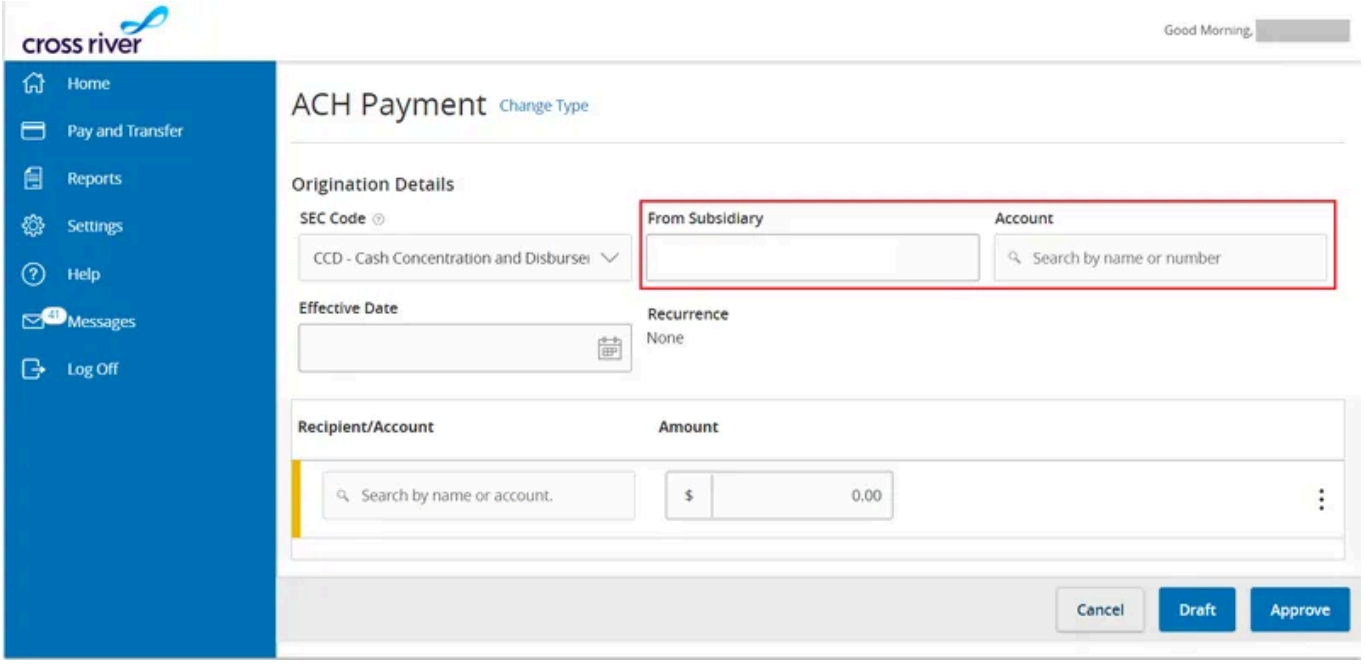
This will bring you to the **Payments Hub**, where you can select a payment type from the list presented.



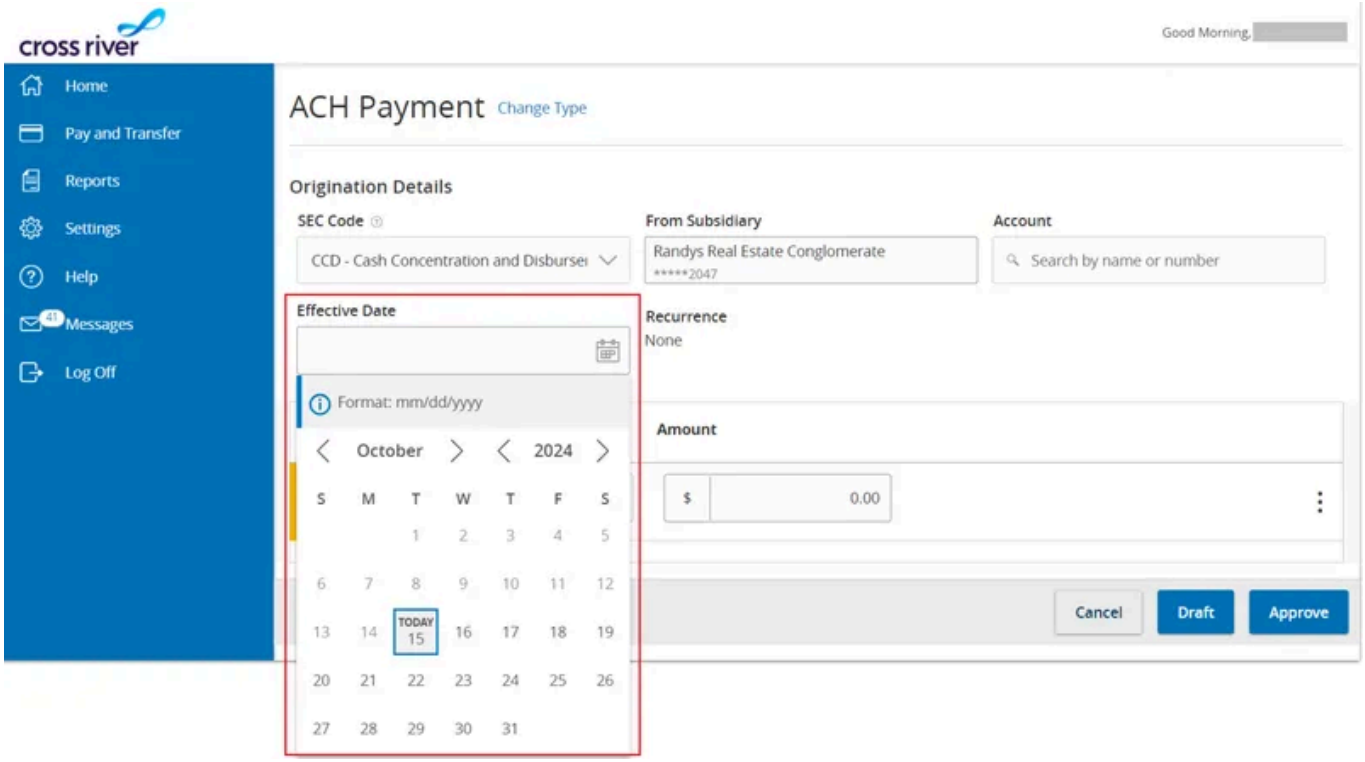
ACH Batch or ACH Payment

By selecting **ACH Batch** or **ACH Payment** from the dropdown menu, you will be shown the necessary fields for originating an ACH push transaction. Batch payments can have mulitple recipients, whereas single payments only have a single recipient.

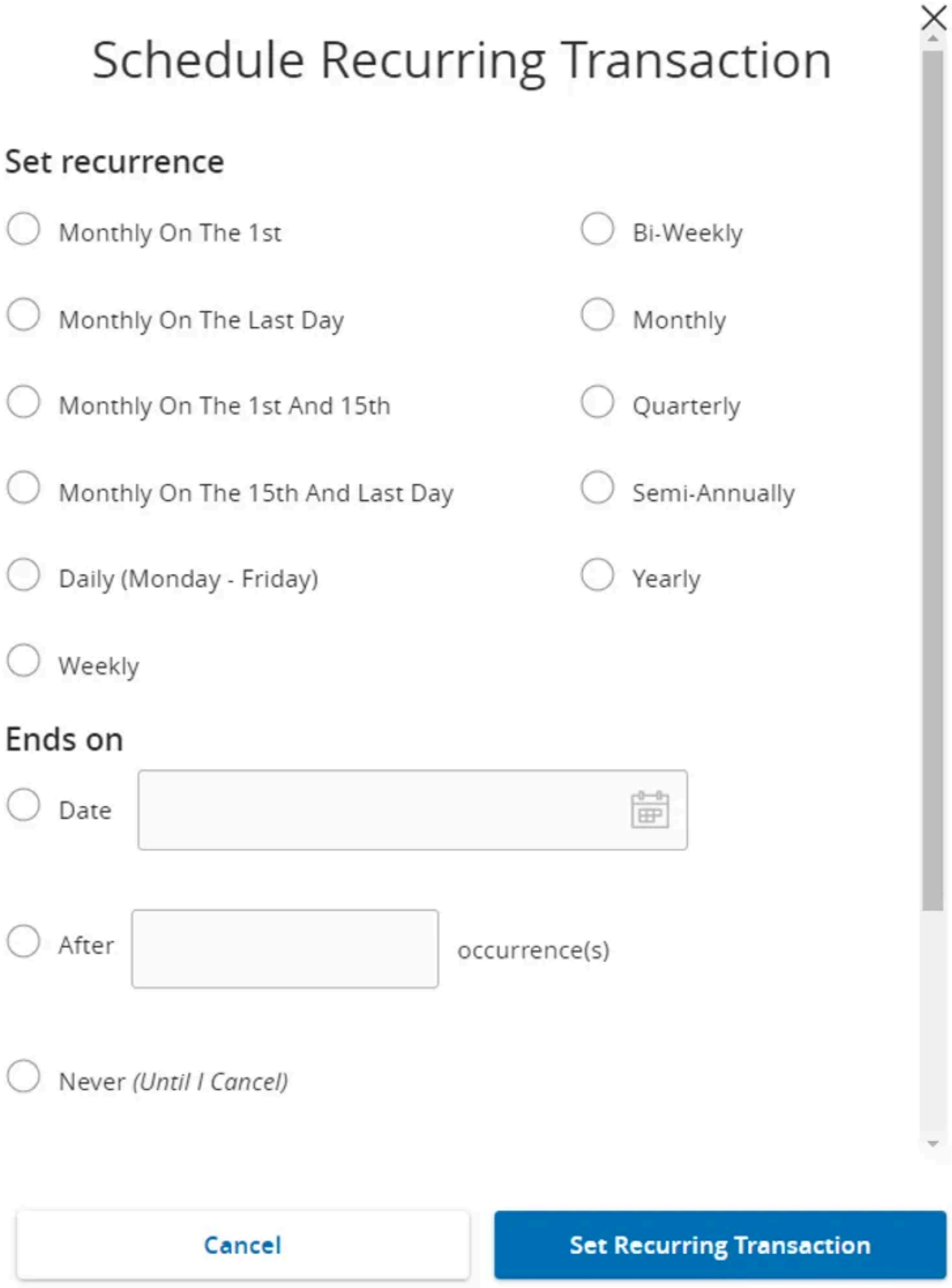
The **SEC Code** will be defaulted to **CCD - Cash Concentration and Disbursement**. You will have to select a **From Subsidiary**, if your company has accounts set up with multiple subsidiaries. Next you will select which specific **Account** you would like the payment to be sent from.



For **Effective Date**, you will select that date that you would like the **ACH Payment** to process. If you select the current date, you are attempting to submit a **Same Day ACH** payment, which is subject to higher fees according to your customer agreement.



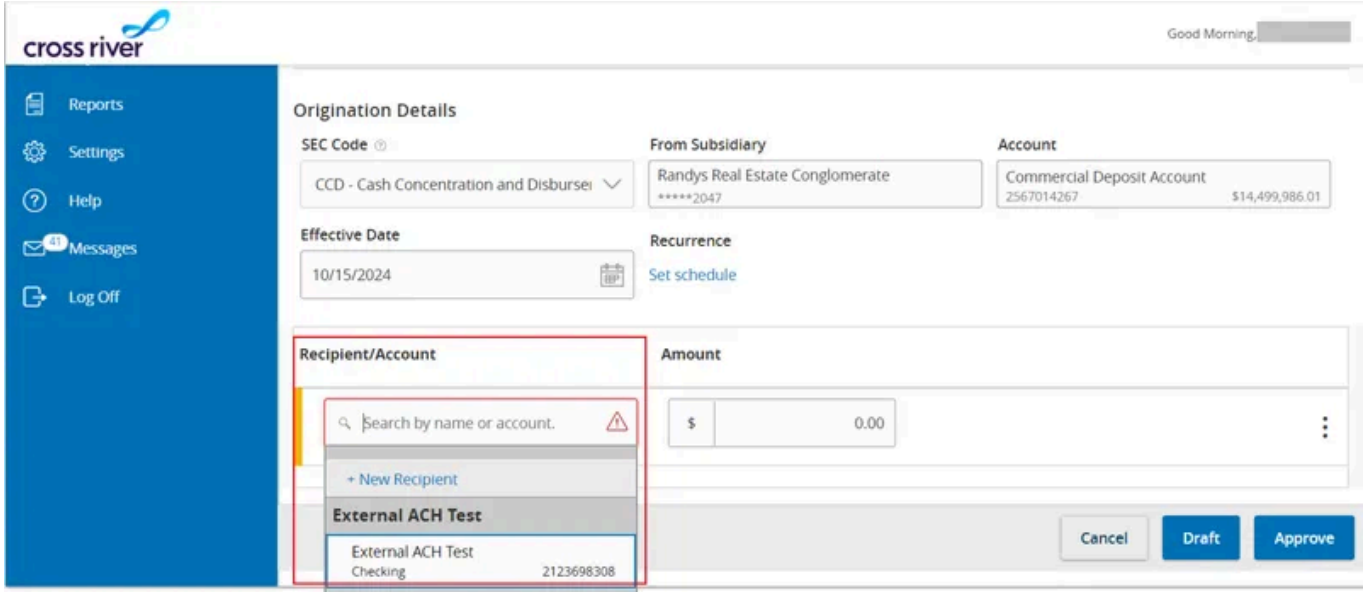
You can also plan a **Recurrence** from this screen, if you would like to schedule a recurring transaction. Here are the options available:



In the **Recipients/Account** field, you need to add the details of the recipient party where you want to send the payment. This field will store any recipient parties you have

previously added, and is searchable.

To add a new or first time recipient see **Manage recipients**.



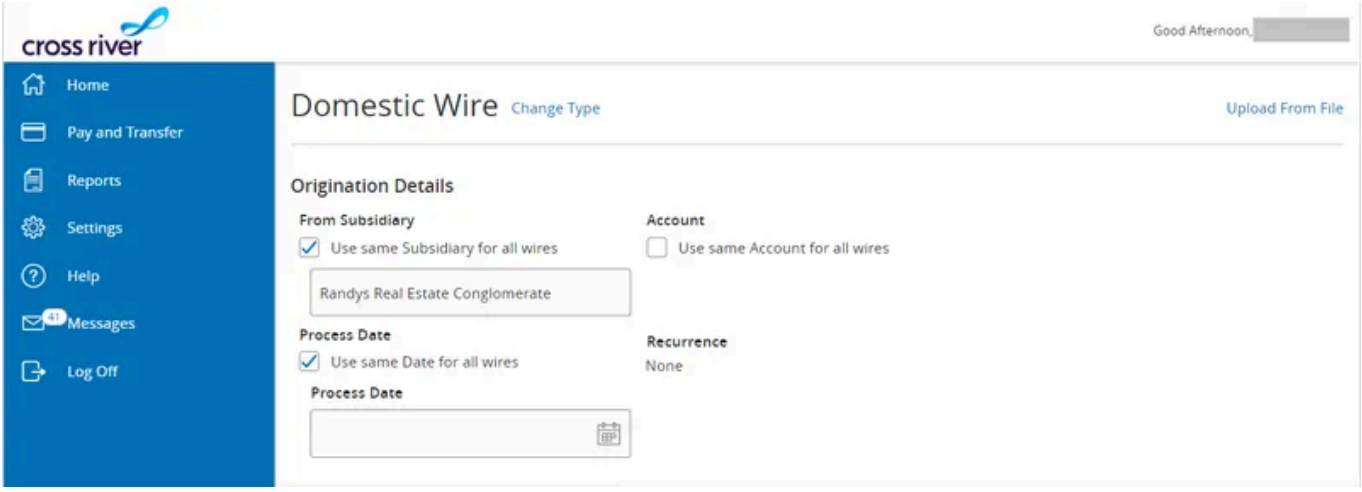
Enter the transaction dollar amount in the **Amount** field.

Finally, if your company has enabled **Dual Action**, you may only be allowed to **Draft** the transaction. The ability to **Draft** or **Approve** transactions is determined by the **User Roles** that your account admin has set up.

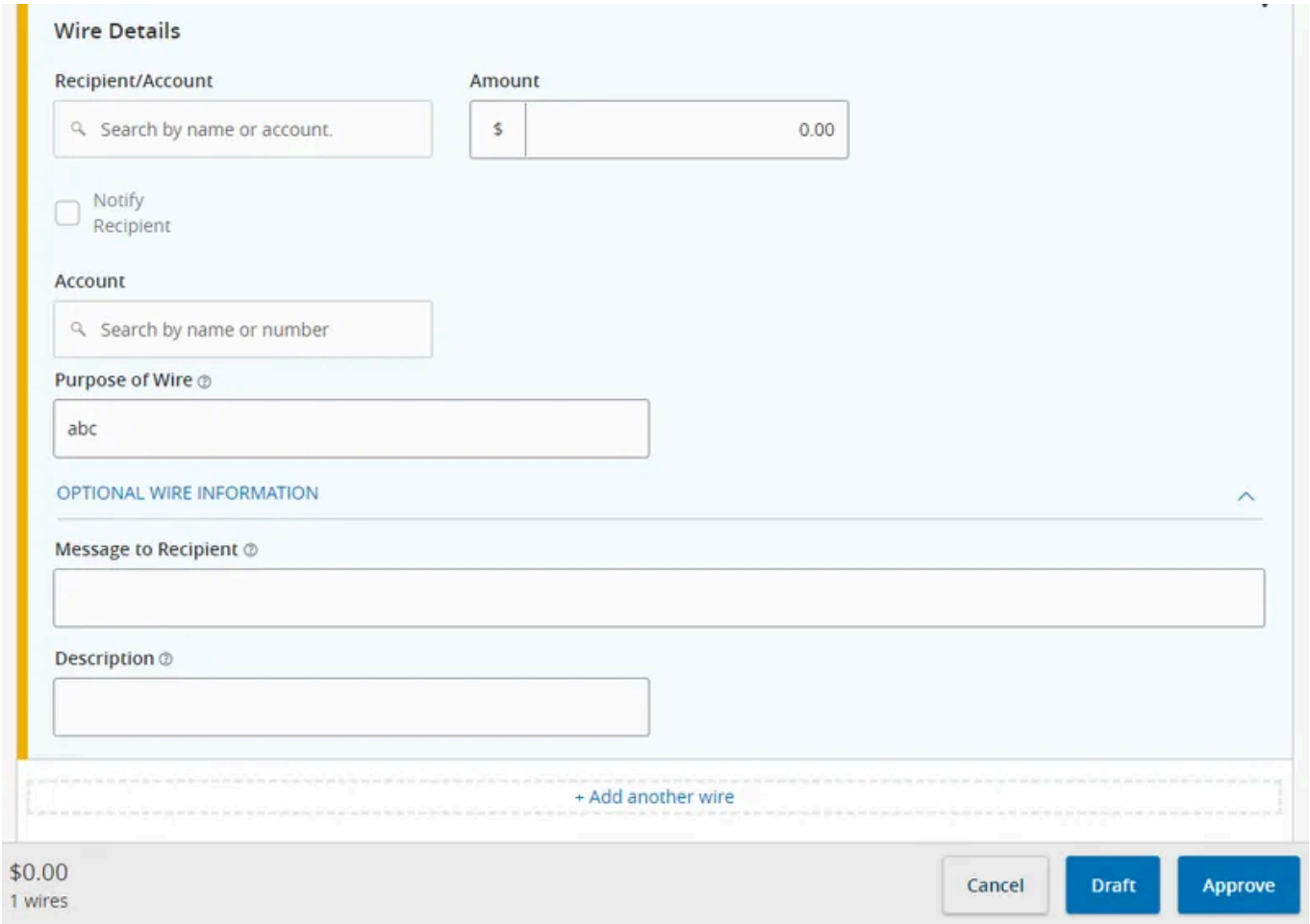
Domestic Wire Payment

By selecting **Domestic Wire** from the dropdown menu, you will be shown the necessary fields for originating a Domestic Wire push transaction.

To start, you will need to fill out the **Origination Details** section, which will include which **From Subsidiary** and **Process Date** should be used for originating the wire.



Next, you will need to fill out the **Wire Details** Section. You will need to input information in the following fields: **Recipient/Account**, **Amount**, **Account**, **Purpose of Wire**.



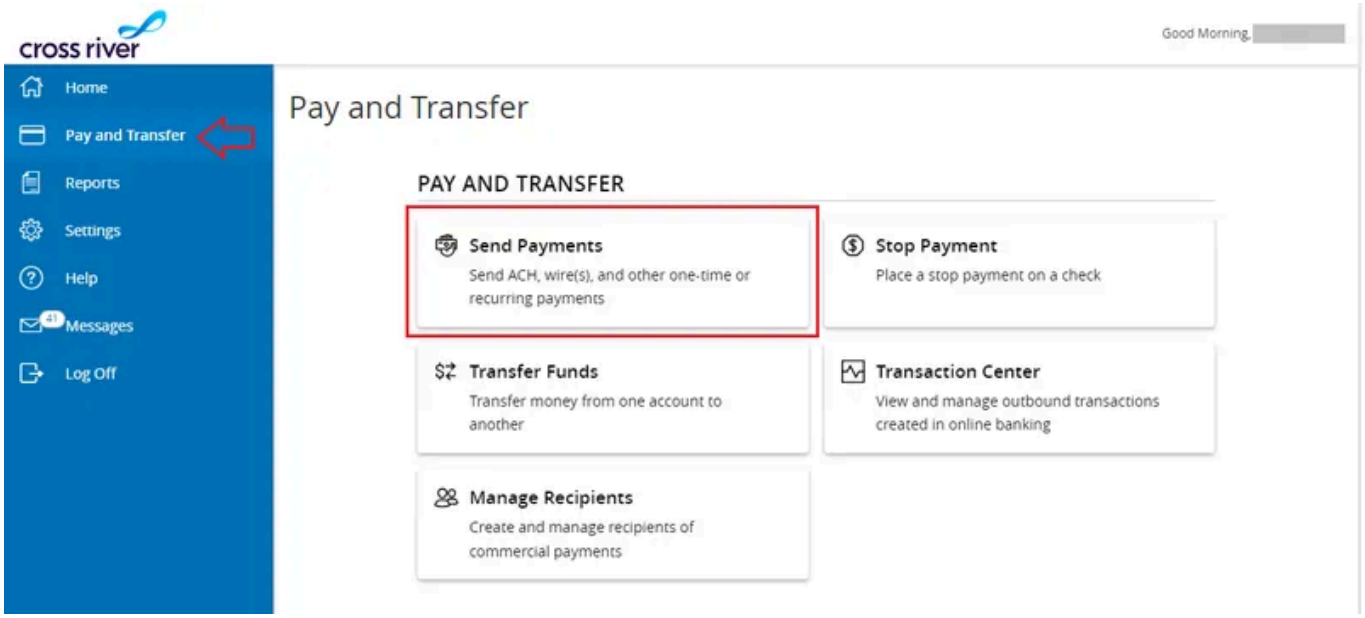
Finally, if your company has enabled **Dual Action**, you may only be allowed to **Draft** the transaction. The ability to **Draft** or **Approve** transactions is determined by the **User Roles** that your account admin has set up.

6.3. Create payment template

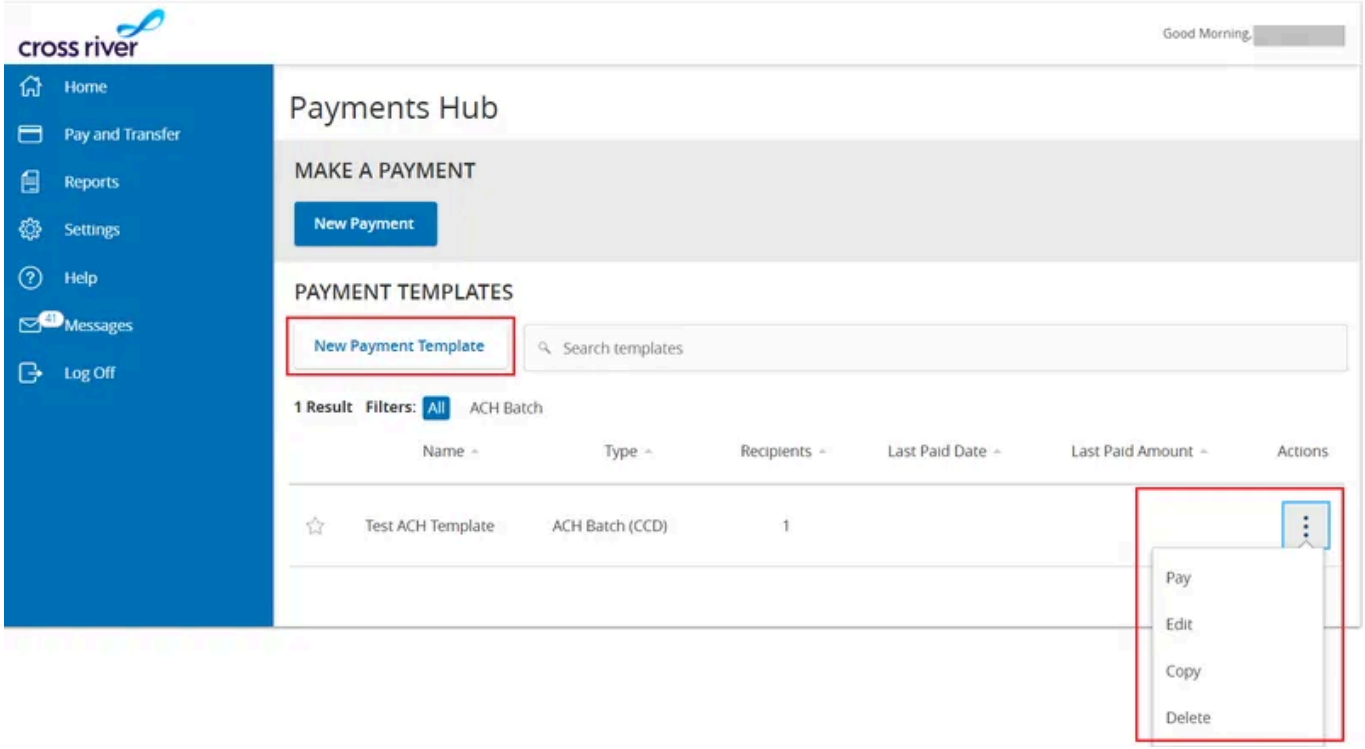
Creating payment templates can help streamline repetitive transactions, saving your company time and reducing the possibility of errors. By setting up templates, users can predefine payment details like recipient information, payment amount, origination details, and payment frequency.

How to Create a Payment Template

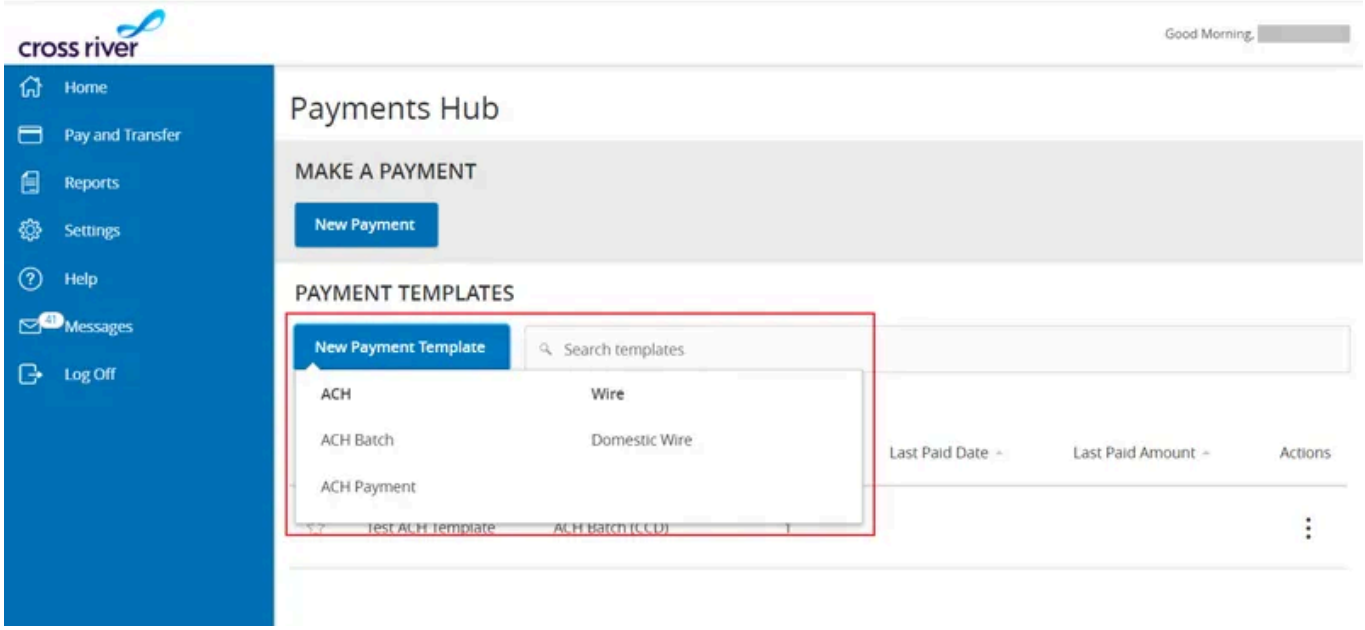
To set up a payment template, navigate to the **Pay and Transfer** tab and click on the **Send Payments** tile.



From here you can create a **New Payment Template**, or edit an existing payment template. You can also originate payments from your existing payment templates on this screen.



If you click on **New Payment Template**, you will have the option to create a payment template for an **ACH Batch**, **ACH Payment**, or **Domestic Wire**.



6.4. 24/7 internal settlement

Description of Payment Type

24/7 Internal Settlement is Cross River's new payment type, which allows customers to settle with each other on our network, 24/7/365. Transactions are processed quickly, within about 15-30 minutes.

To use this payment type, both the originator and the receiver must hold a deposit account with Cross River. This requires a configuration on the product by a Cross River employee, so please reach out to your Relationship Manager if you are unclear if this is enabled on your account.

How does it work?

This transaction type leverages ACH payment formatting, but Cross River will not send these transactions out to an ACH Operator. Functionally, these transactions are secure account to account transfers, if they are between two parties on our network.

Pricing

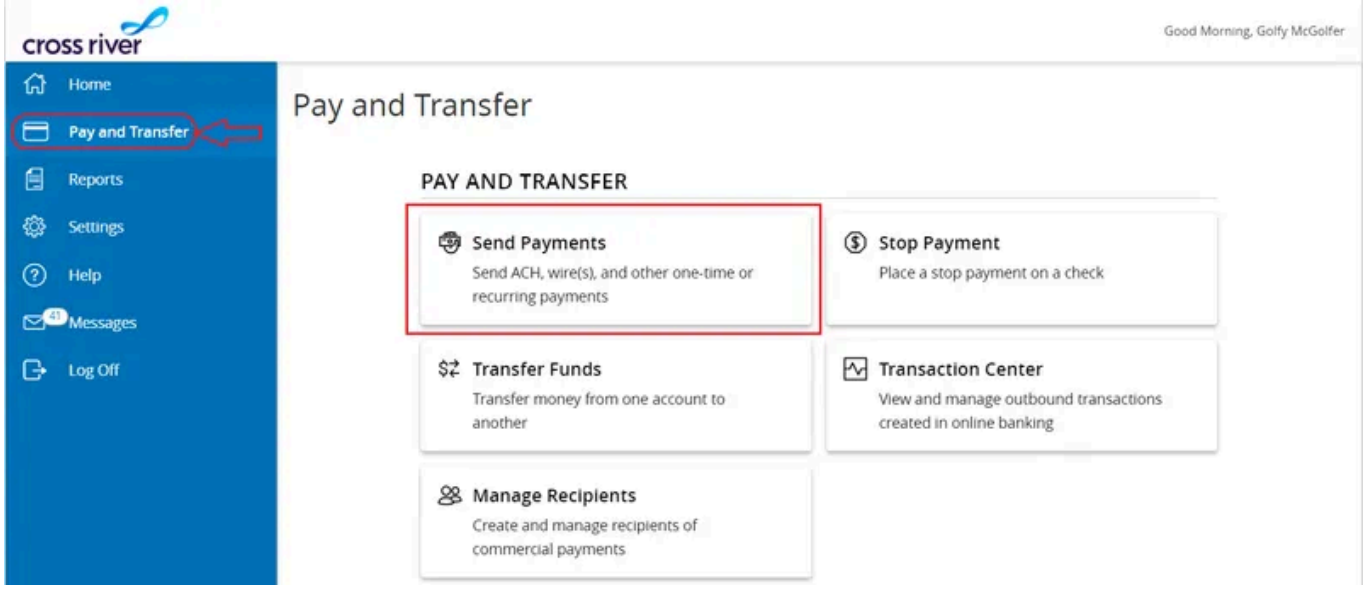
Standard ACH transaction fees apply to **24/7 Internal Settlement** transactions. Refer to your Customer Agreement to see the fee schedule that applies.

Guide for 24/7 internal settlement transactions

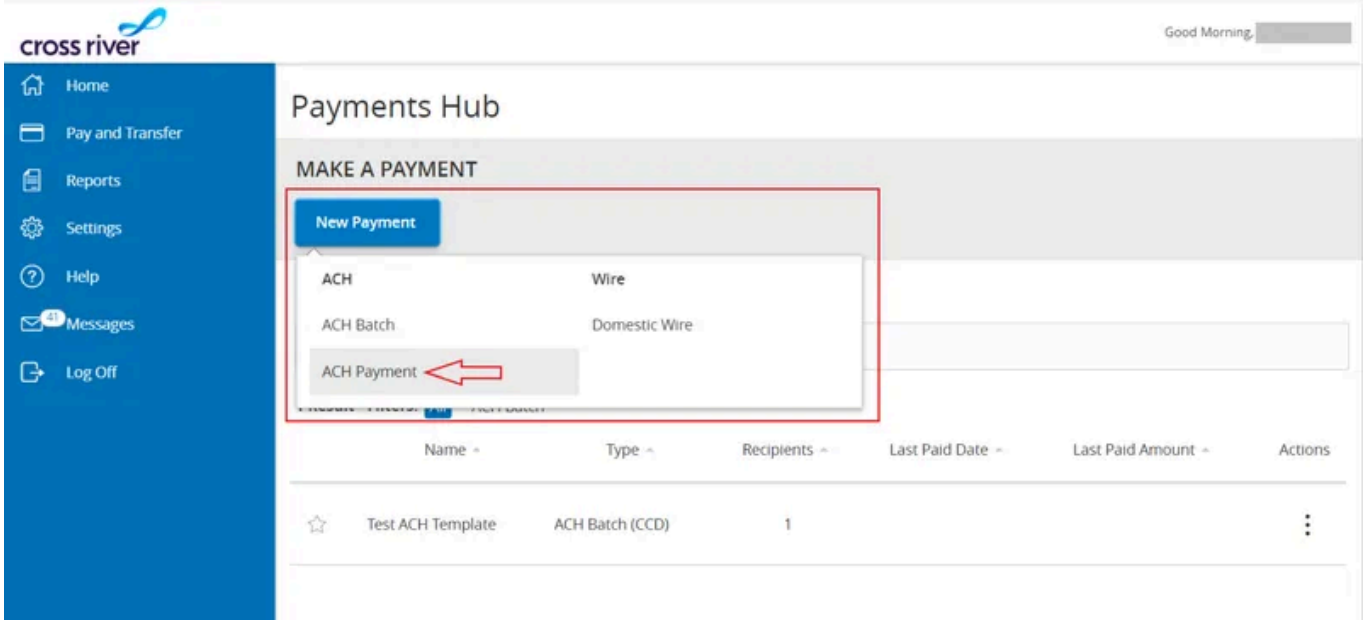
To make this payment type, the payment originator and recipient will need a **Settlement Account** at Cross River. Here are the steps to submit a **24/7 Internal Settlement** transaction.

Navigating to Transaction Type

Navigate to the **Pay and Transfer** tab and click on **Send Payments**.

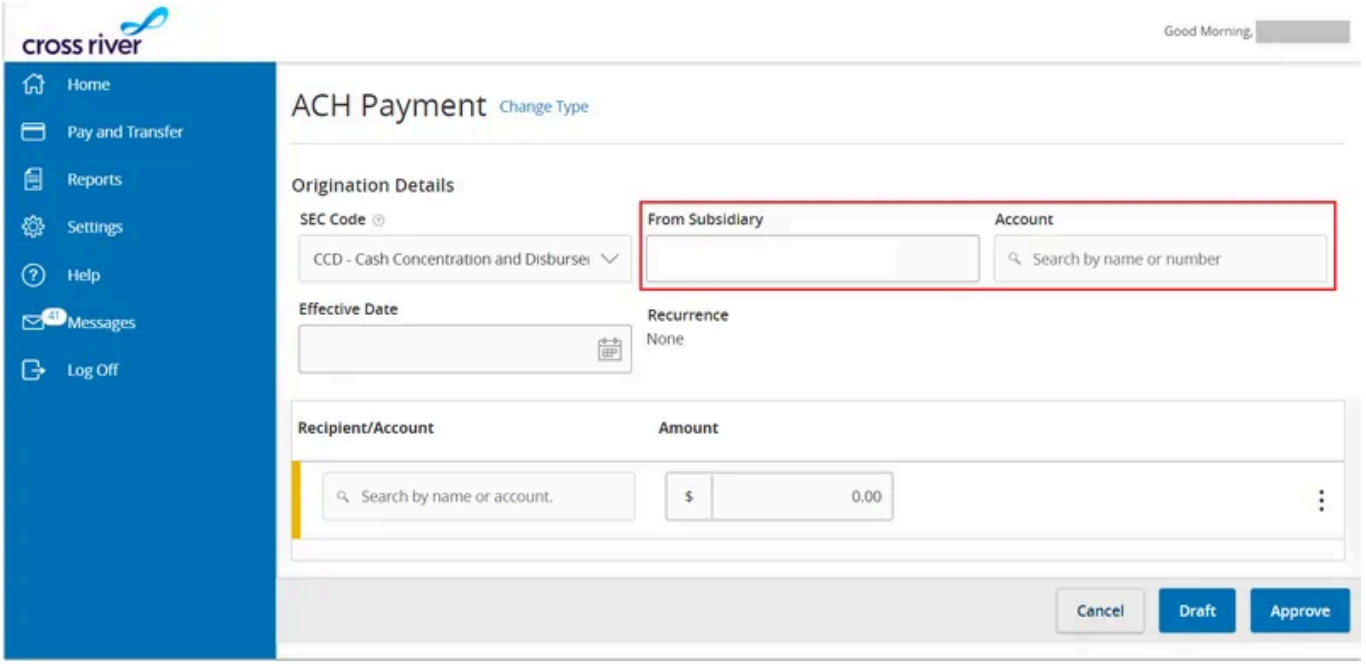


Click on **New Payment** and then **ACH Payment**.

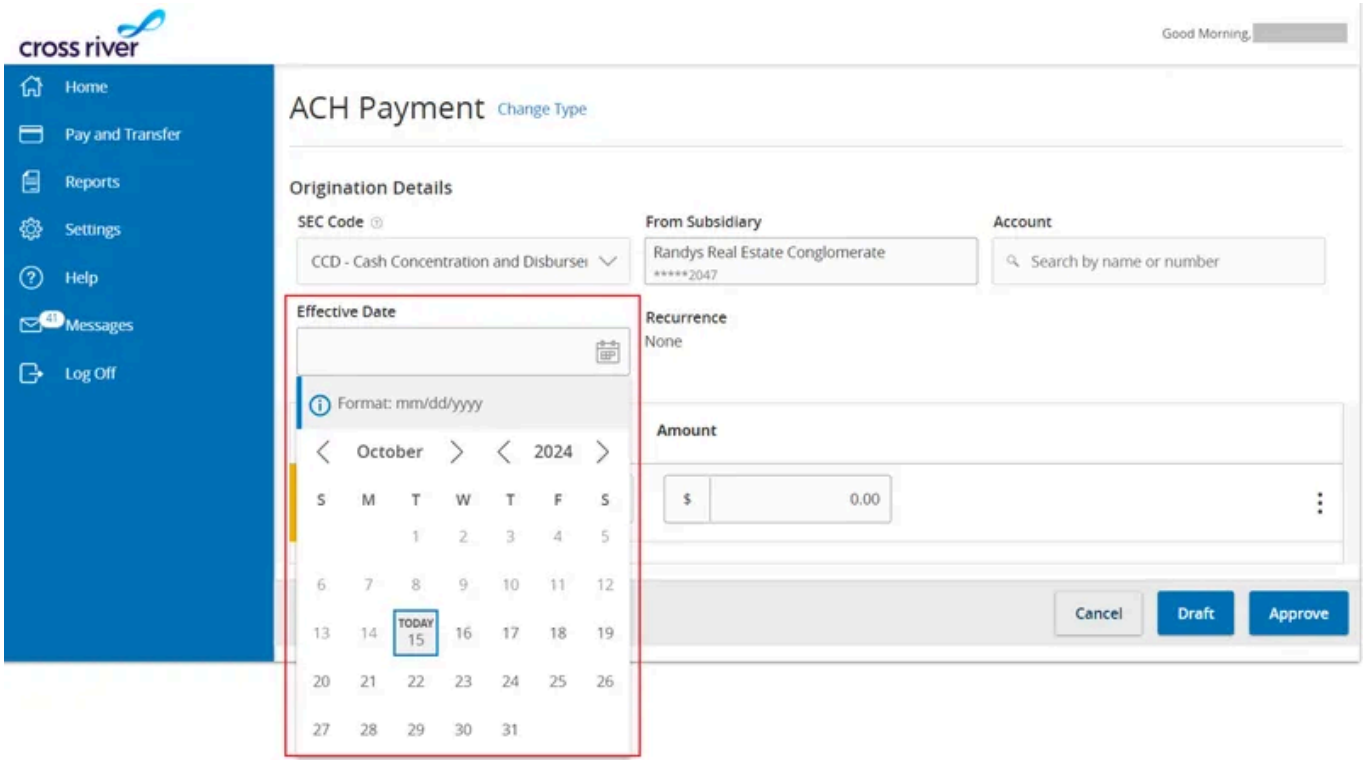


Originating Party Fields

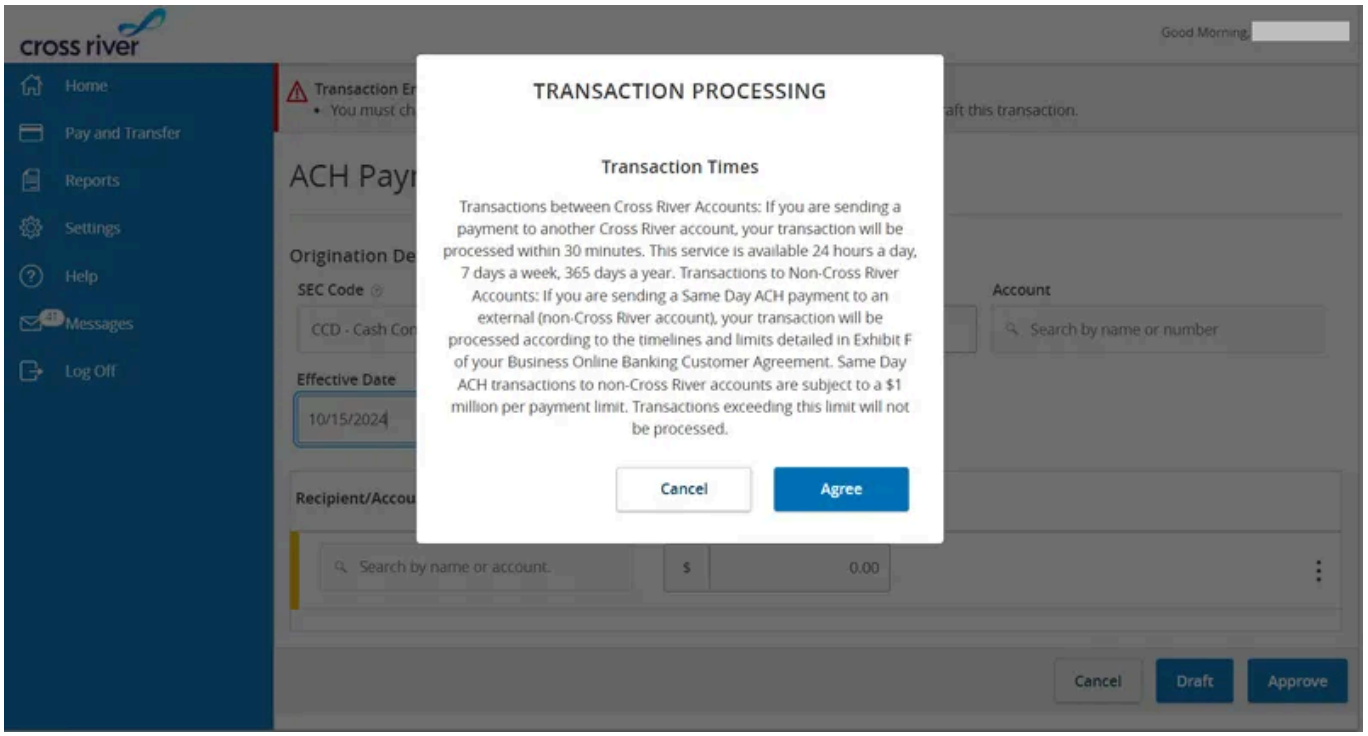
The **SEC Code** should be defaulted to **CCD - Cash Concentration and Disbursement**. You will have to select a **From Subsidiary**, if your company has accounts set up with multiple subsidiaries. Next you will select which specific **Account** you would like the payment to be sent from.



For **Effective Date**, you need to select the current date on the calendar pop-up. DO NOT plan this payment type for a future date, as this will always process within 15 - 30 minutes of the transaction submission.



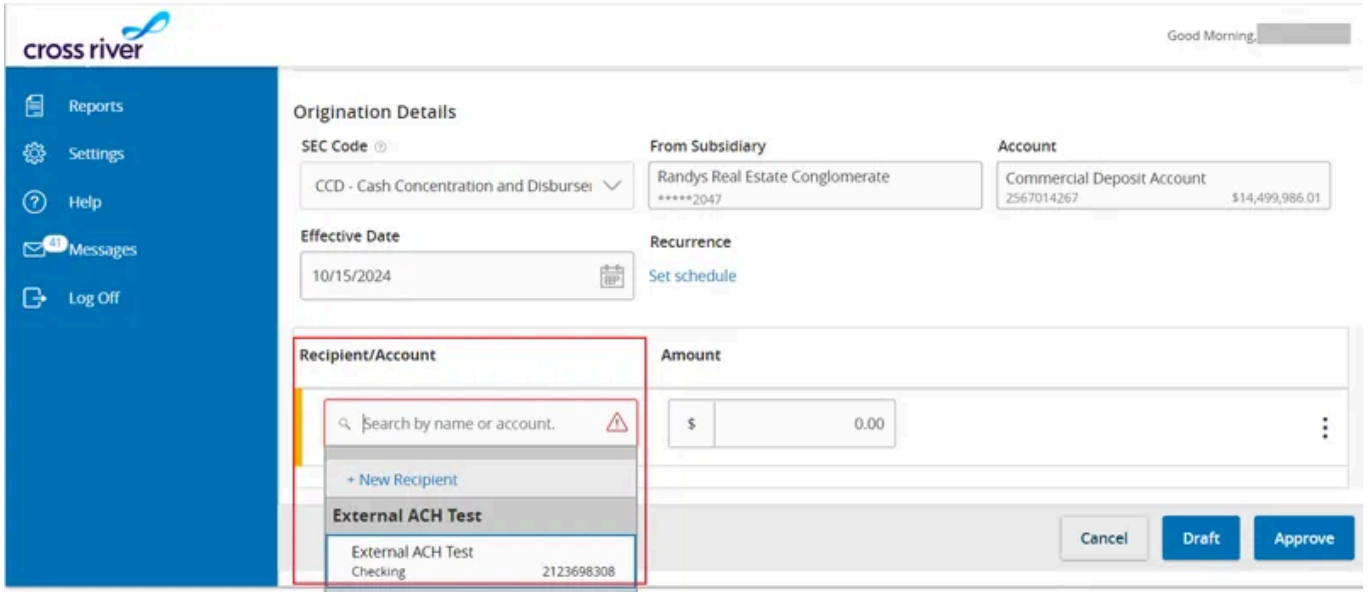
When you select the current date in the **Effective Date** field, the following disclosure displays. You should read the disclosure to ensure your understanding, and need to click **Agree** in order to continue with this payment type.



Recipient Party Fields and Submission

In the **Recipients/Account** field, you need to add the details of the recipient party where you want to send the payment. This field will store any recipient parties you have previously added, and is searchable.

To add a new or first time recipient refer to **Manage recipients**.



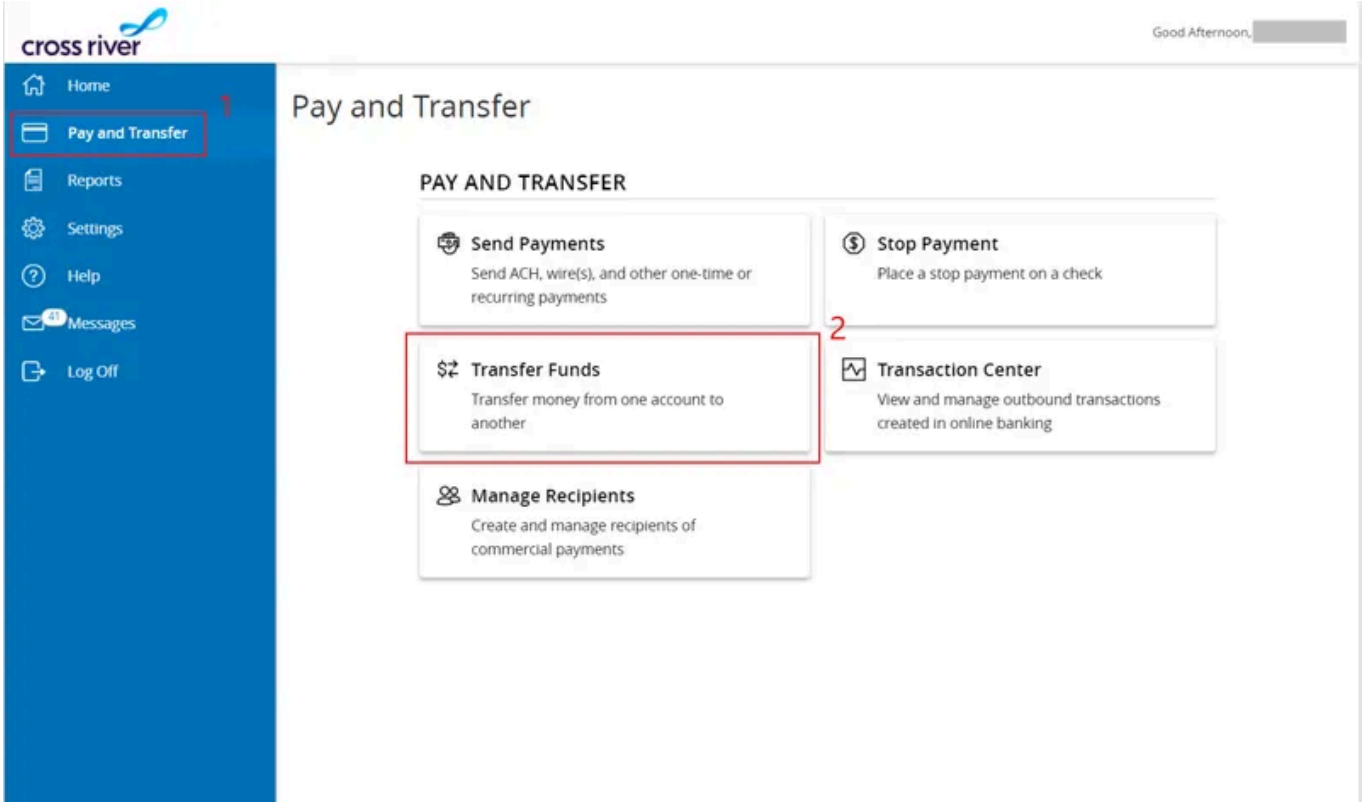
Enter the transaction dollar amount in the **Amount** field.

For this transaction type the payment can be greater than $1M, since the transaction will never leave Cross River. Same Day ACH payments for over $1M, made to recipient accounts outside of Cross River, will fail.

Finally, if your company has enabled **Dual Action**, you may only be allowed to **Draft** the transaction. The ability to **Draft** or **Approve** transactions is determined by the **User Roles** that your account admin has set up.

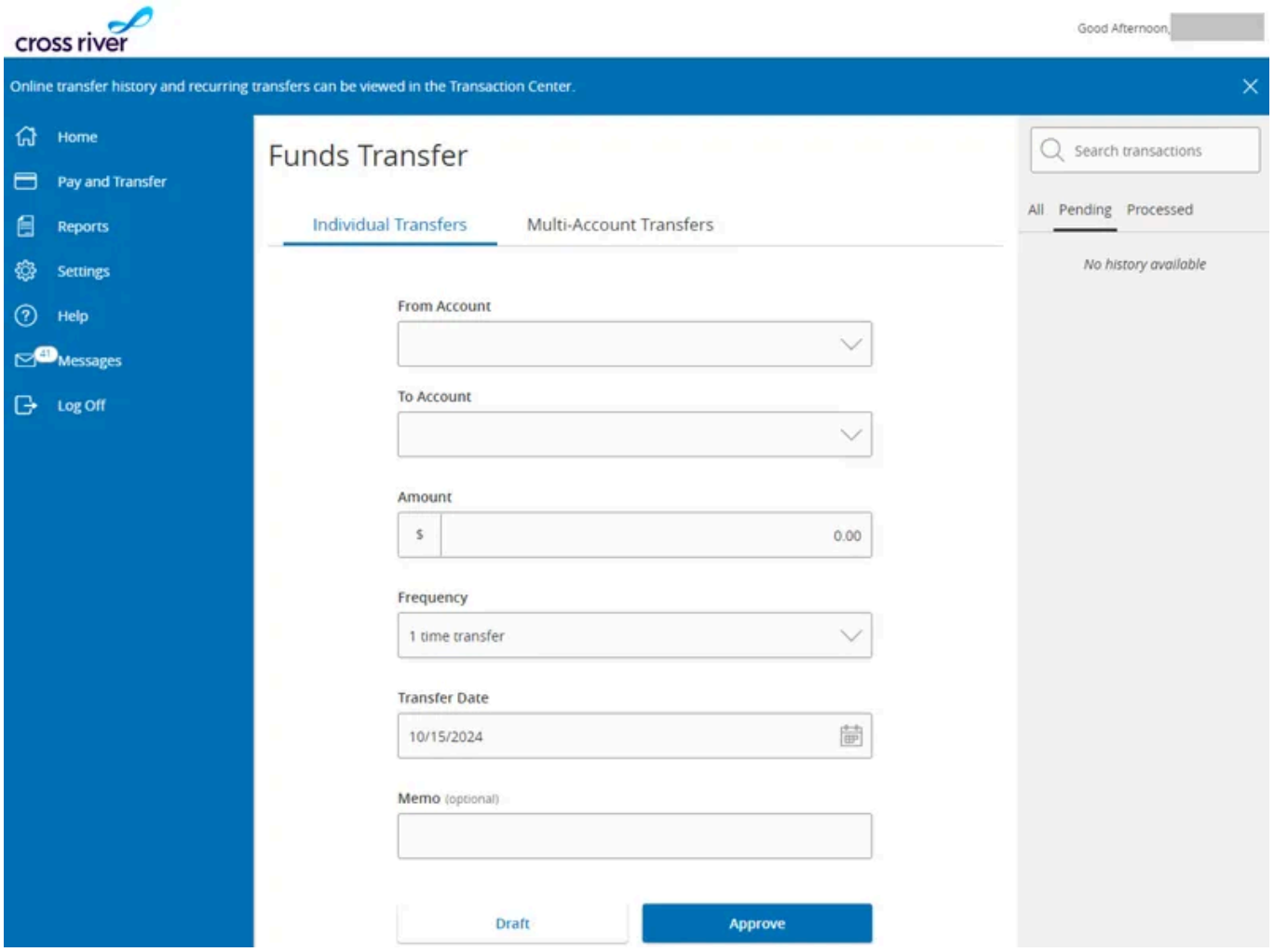
6.5. Transfer funds

You can easily move money between your various Cross River accounts on the digital banking platform. To move money between your internal accounts, go to the **Pay and Transfer** tab, and click **Transfer Funds**.



Individual Transfers

Here you can set up your transfer and select the **Amount**, **Frequency**, and **Transfer Date**.



To complete a transfer, you will then select either **Draft** or **Approve**. If you have **Dual Action** enabled, then you can only select **Draft**, and another account user will have to approve the transaction.

If your account admin has requested that transfers require **dual authorization** (a.k.a 2FA), you will be sent a **Secure Access Code** by email or text message.

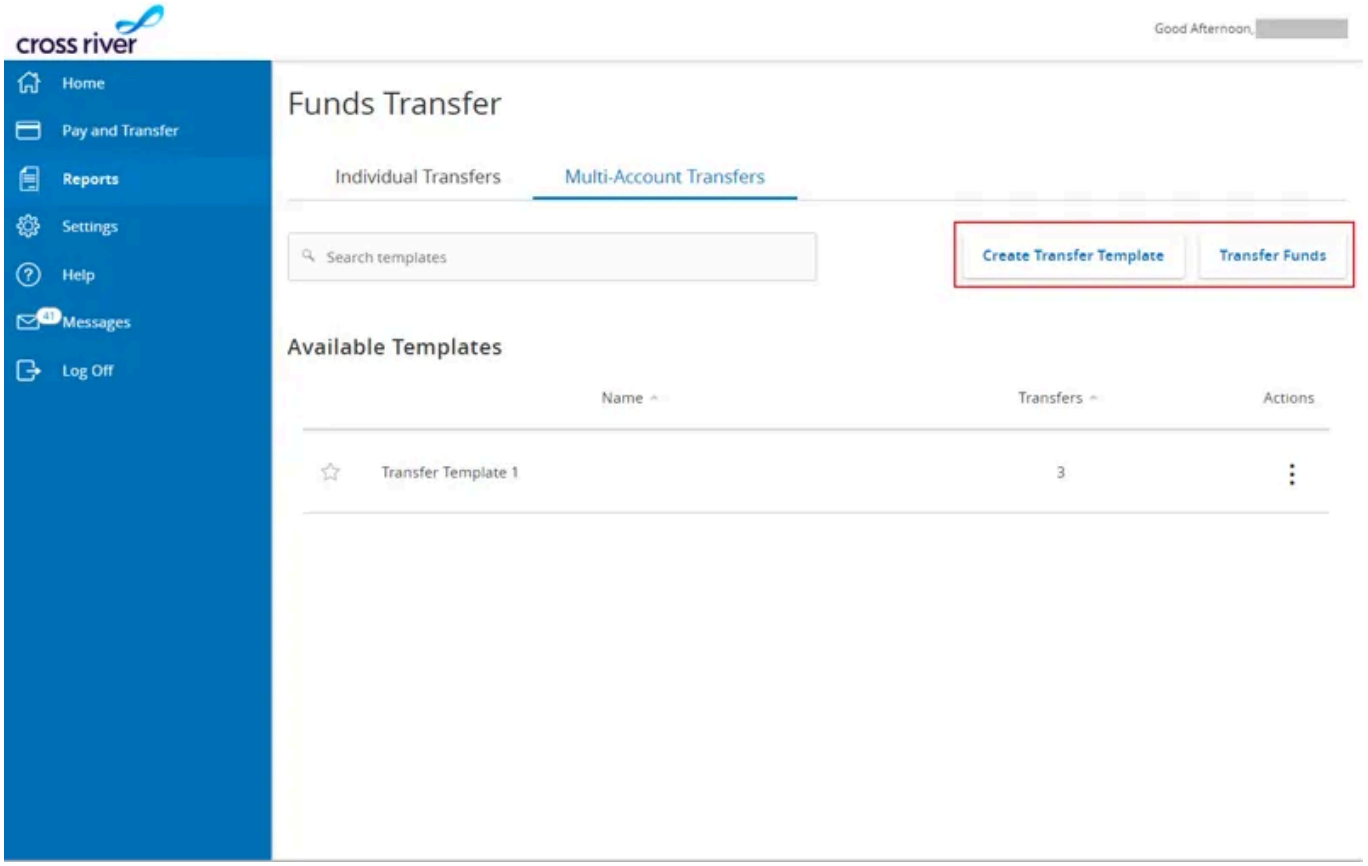
Multi-Account Transfers and Transfer Templates

Multi-Account Transfers

To move money between multiple internal accounts, and to create templates of common internal transfers, select the **Multi-Account Transfers** tab.

To initiate a one-time multi-account transfer, select the **Transfer Funds** box and follow the instructions to select your transfer.

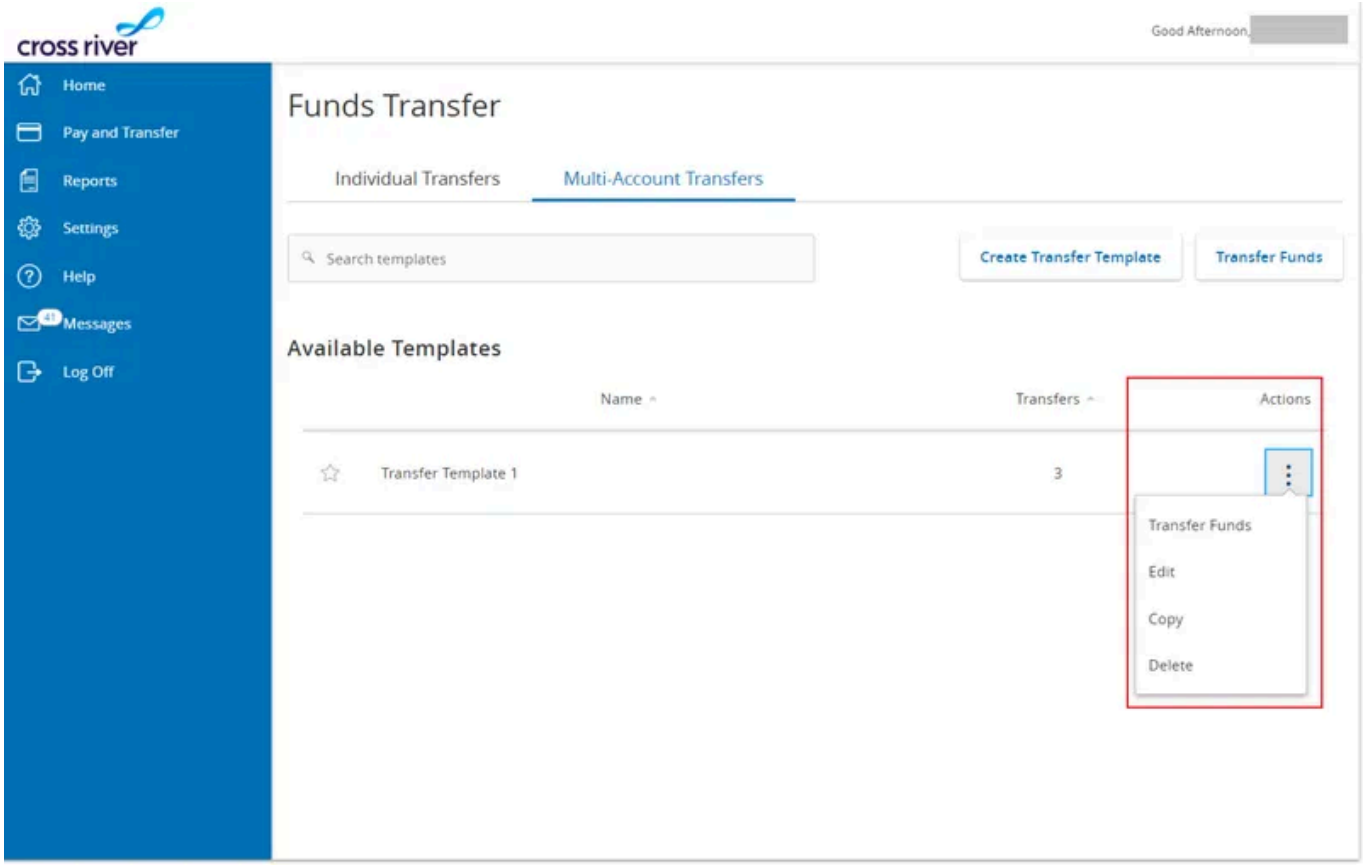
The same **Draft** and **Approve** options are available as previously described.



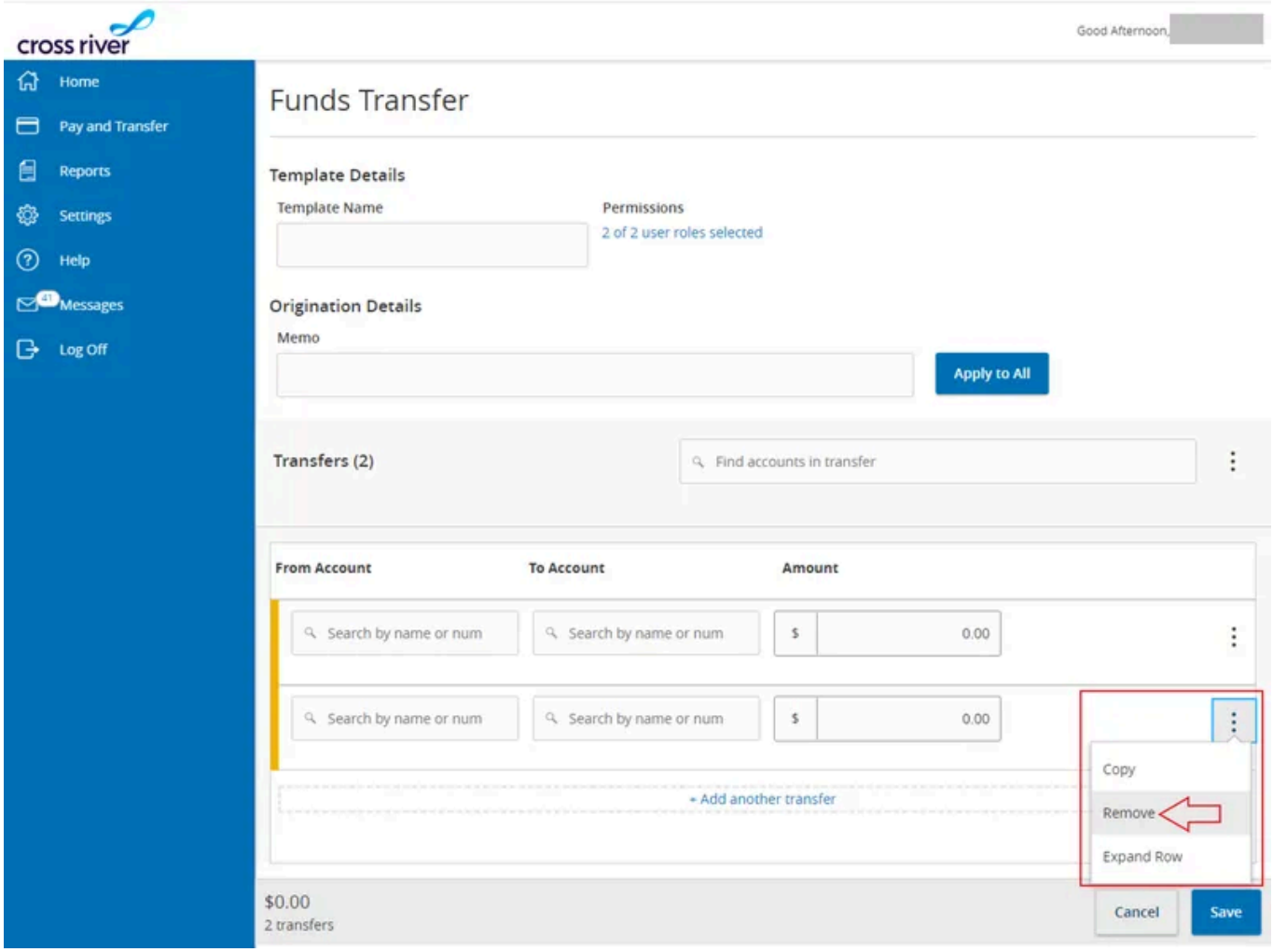
All saved templates will be on this page, including a search function if you have many templates.

Transfer Templates

To action a saved template, click the 3-dot menu next to the one you want to manage.

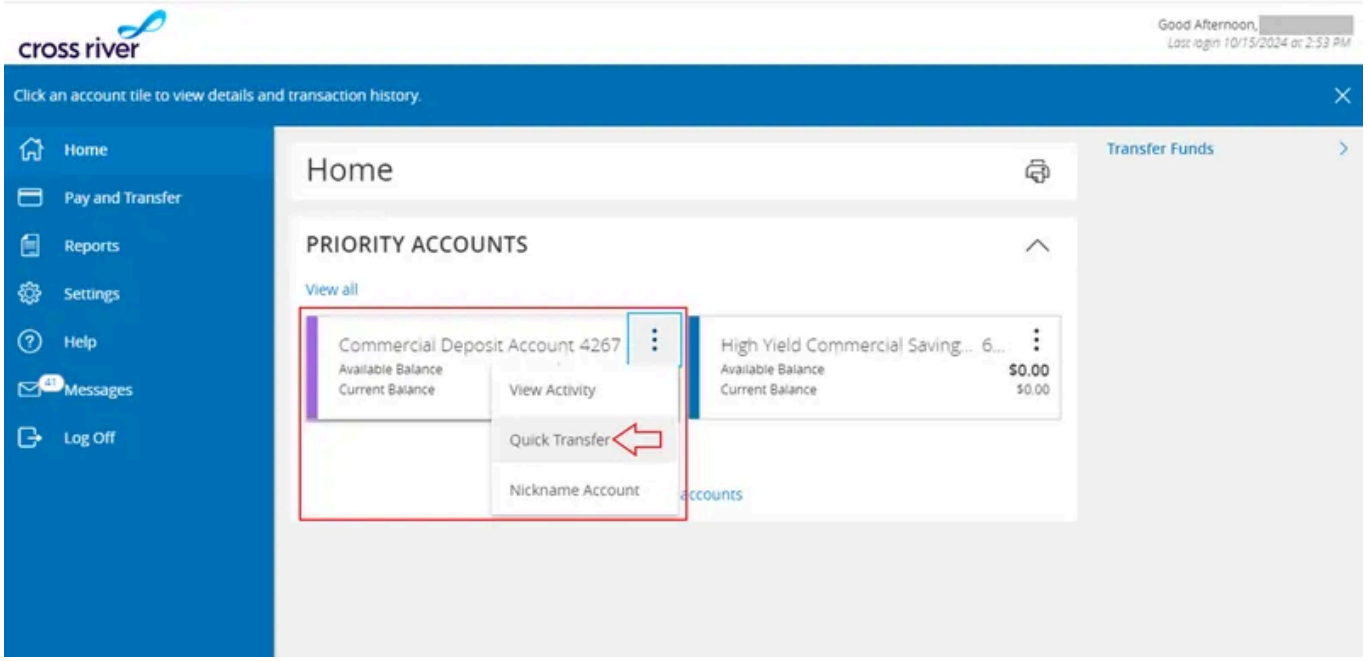


To create a template for a single account transfer (a transfer between 2 different accounts), select the **Create Transfer Template** button, and **Remove** the rows until there is only one row remaining. Remember to click **Save**.



The Quick Transfer option

On the **Home** tab, you have the ability to transfer funds between your own accounts that are displayed on that page. Click on the 3-dot menu next to the account you wish to leverage as the **From Account**, and then click **Quick Transfer**.



This will bring up the **Quick Transfer** screen, where you can easily submit an internal transfer between your own accounts.

Quick Transfer

From Account

Commercial Deposit Account 2567014267 $14,499,986.01

To Account

Amount

$ 0.00

Transfer Date

10/15/2024

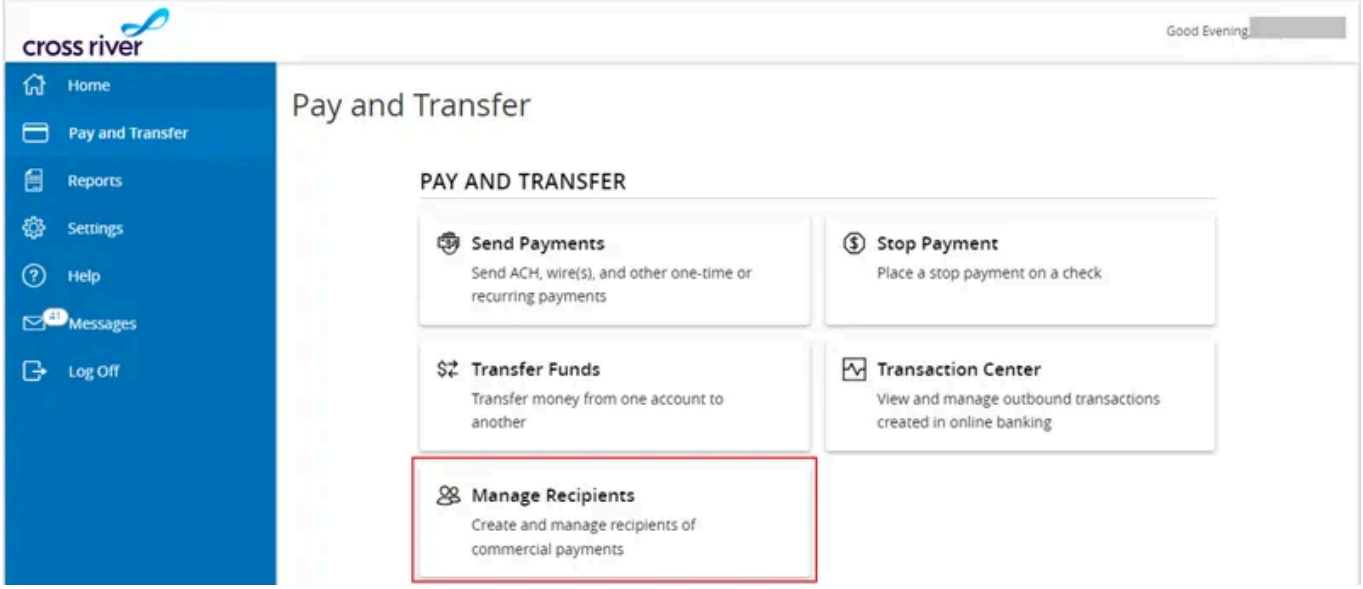
Advanced Options | Draft | Approve

6.6. Manage recipients

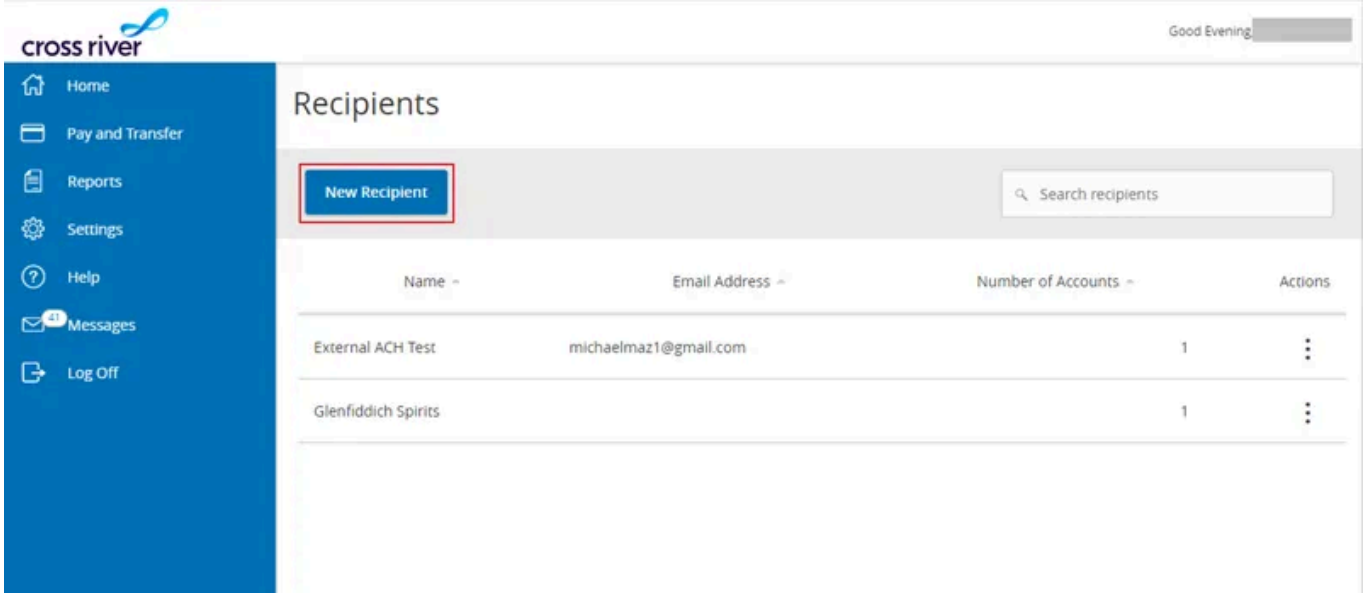
Adding payment recipients in advance helps improve efficiency for your company during the payment process. This feature allows you to quickly add and save trusted recipients, ensuring you accurately send funds to the right individuals or businesses in the future. Explore below how to set up and manage your recipients effectively for a smoother banking experience.

Navigating to Manage Recipients

First navigate to the **Pay and Transfer** tab, and click on the **Manage Recipients** tile.



Next, click on the **New Recipient** button, in order to be brought to the **Add Recipient** screen.



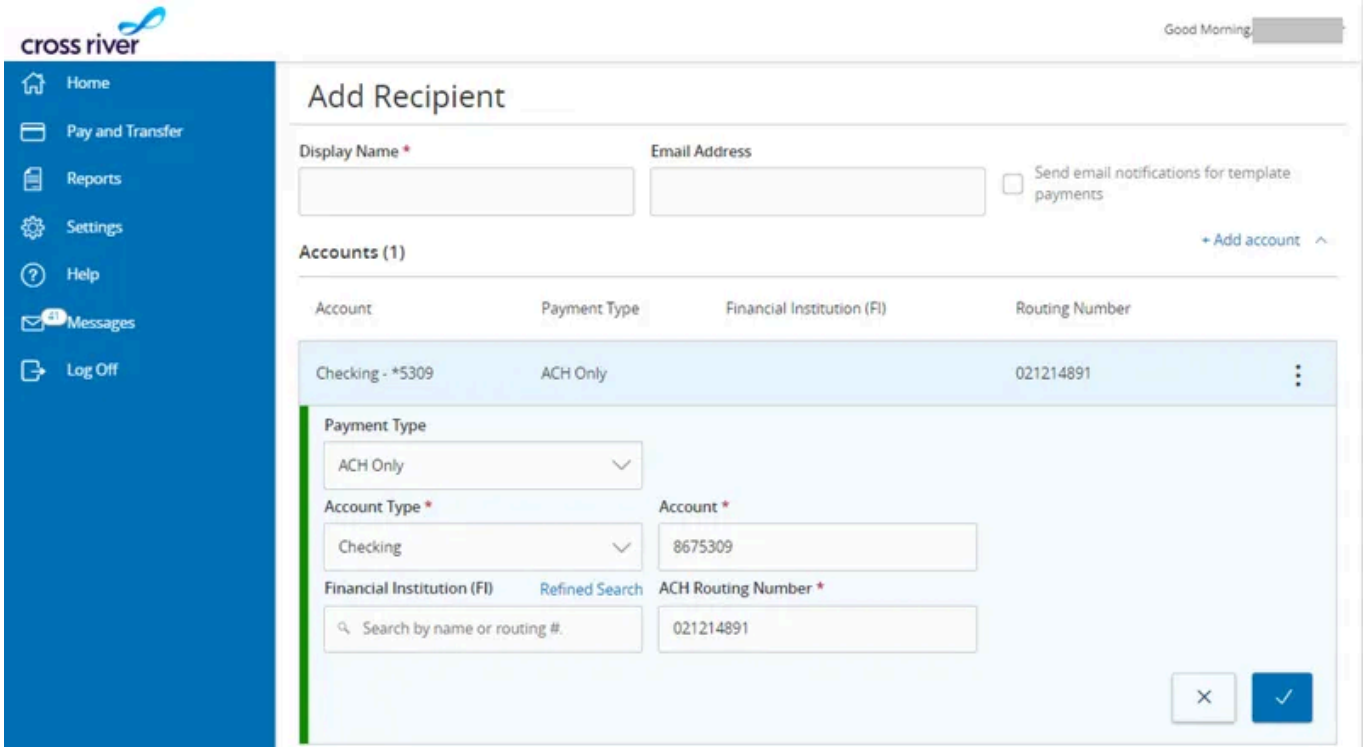
The information you will need to provide for an ACH vs. a Wire recipient will differ, since these payment types differ greatly.

Adding an ACH Recipient

You will first have to enter information in the top two fields:

	Description
Display Name	The name of the account that will be stored in your online banking portal for future use.
Email Address	Add your email address and click the box if you wan to send email notification for template payments

Of these two fields, only **Display Name** is mandatory since this will be how the recipient name is stored in the digital banking platform, for future payments.



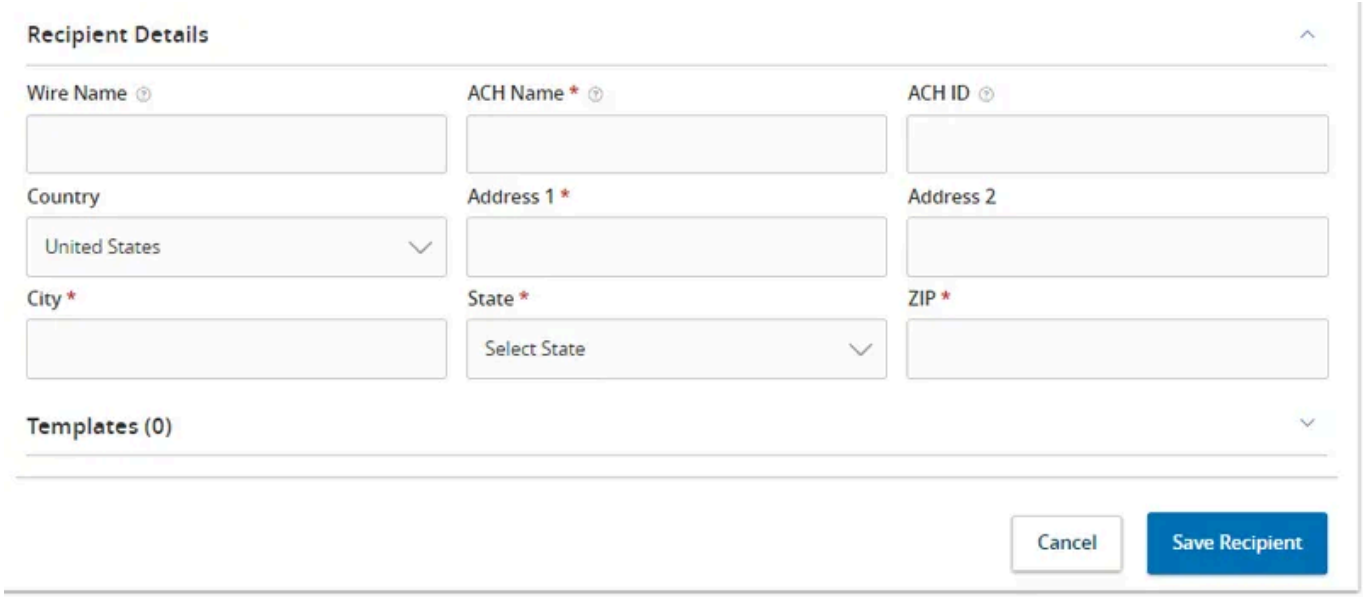
When you proceed to select **ACH Only** as the **Payment Type**, the following fields will display which are all required:

Field	Description
Account Type	Select one. **Checking**, **Savings**, and **Loan**.
Account	This is the account number of the recipient.
ACH Routing Number	This is the routing number of the recipient's bank.

Click on the blue check mark when you are done filling out this section.

Next, you'll need to fill out the **Recipient Details** section. The following fields are relevant in this section:

Field	Details
ACH Name	Required field where you enter the recipients name (maximum of 22 characters)
ACH ID	Optional field that will be autopopulated during payment submission, so leave this empty (maximum of 15 characters)
Address 1-2, City State and ZIP	Recipients full address including City, State and ZIP code



Lastly, click on the **Save Recipient** button to ensure that the recipient is stored in the digital banking platform.

Adding a Wire Recipient

Once you've navigated to the **Add Recipient** screen, added a **Display Name**, and selected **Wire Only** as the **Payment Type**, the following fields will display:

Payment Type
Wire Only
Recipient Type
Domestic
Account *
Financial Institution (FI)
Refined Search
Search by name or routing #.
Recipient FI
Name *
Country *
United States
FI ABA Number *
Address 1 *
Address 2
City *
State *
Select State
Postal Code *
Intermediary FI
Name
Country
United States
Wire Routing Number
Address 1
Address 2
City
State
Select State
Postal Code

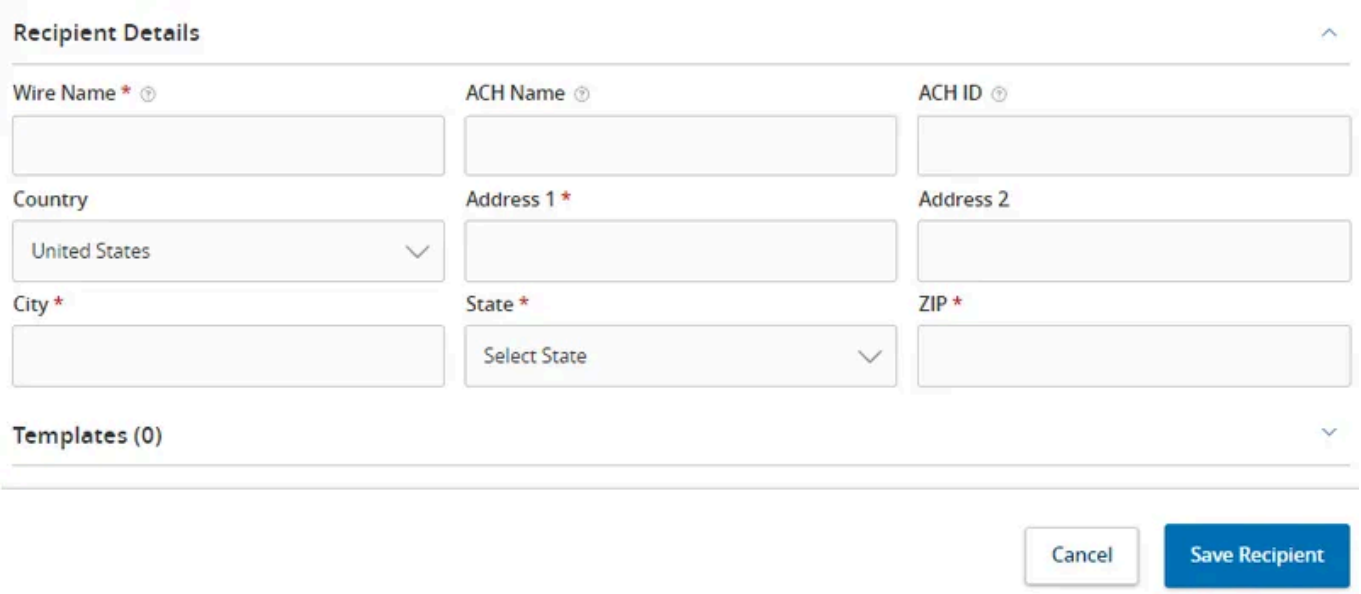
These fields are required in this section:

Field	Description
Account	This is the account number of the recipient.
Recipient FI Name	The name of the bank where the recipient's account is.
FI ABA Number	American Bankers Association Routing Number for the recipient's bank.
Address 1-2, City State and Postal Code	Recipient bank's full address including City, State and Postal code

Click on the blue check mark when you are done filling out this section.

Next, you'll need to fill out the **Recipient Details** section. The following fields are relevant in this section:

Field	Details
Wire Name	Required field where you enter the recipients name, as recognized by Recipient's FI (maximum of 22 characters)
Address 1-2, City State and ZIP	Recipients full address including City, State and ZIP code



Lastly, click on the **Save Recipient** button to ensure that the recipient is stored in the digital banking platform.

6.7. High yield commercial savings

The **High Yield Commercial Savings Account** is a new type of deposit account within Digital Banking. It is an interest-bearing account that leverages a **tiered rate card**. You will earn higher yield, the more money you put into an account.

You *must* have a **Commercial Deposit Account** with Cross River to open **High Yield Commercial Savings Account**. This account has no outbound transaction features, you can only transfer funds to your **Commercial Deposit Account** within the Digital Banking portal. From your **Commercial Deposit Account**, you will be able to transact with external accounts. The minimum opening balance to earn interest is $2million.

Pricing

No additional fees apply this account type. The maintenance fee for the **Commercial Deposit Account** applies across the total balance that a customer has in all Cross River Digital Banking accounts. No outbound transactions are allowed out of this account (aside from internal book transfers) so no additional fees apply for this account type.

How does the account work?

The **High Yield Commercial Savings Account** is configured under CRB Direct Digital Banking.

- Interest is compounded daily and is credited to the account monthly.
- If you close your account before interest is credited, you will not receive the accrued interest.
- The daily balance method is used to calculate interest on the account.
- This method applies a daily periodic rate to the principal in the account each day.

There are no limits to the number of online transfers between the **High Yield Commercial Savings Account** and your other deposit accounts at Cross River.

Transfers directly from a **High Yield Commercial Savings Account** to accounts at a third-party institution are not possible. To transfer funds to an external account, you must first move the money into your **Commercial Deposit Account**.

Current rate table as of April 2024 (subject to change)

Tier	Minimum balance to obtain Annual Percentage Yield (APY) $	Interest rates %	APY %
1	2,000,000-4,999,999	3.2	3.25
2	5,000,000-9,999,999	3.92	4.00
3	10,000,000-24,999,999	4.16	4.25
4	25,000,000-49,999,999	4.40	4.5
5	50,000,000+	4.64	4.75

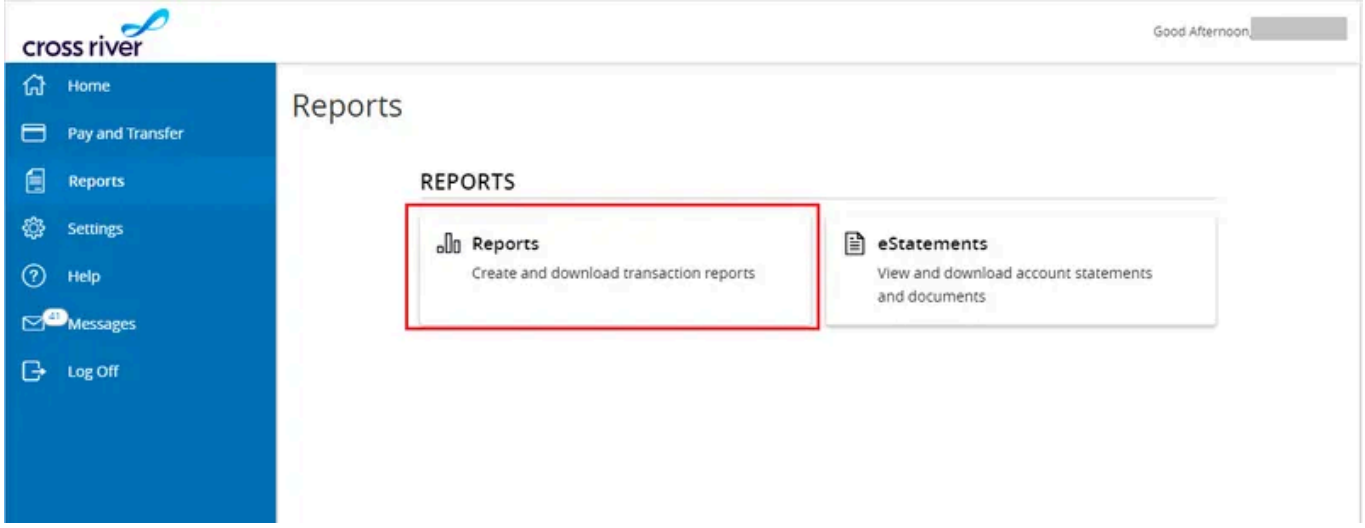
6.8. Reports

Within the Cross River digital banking platform, you will be able to pull various types of account reporting. To access this reporting, navigate to the **Reports** tab.

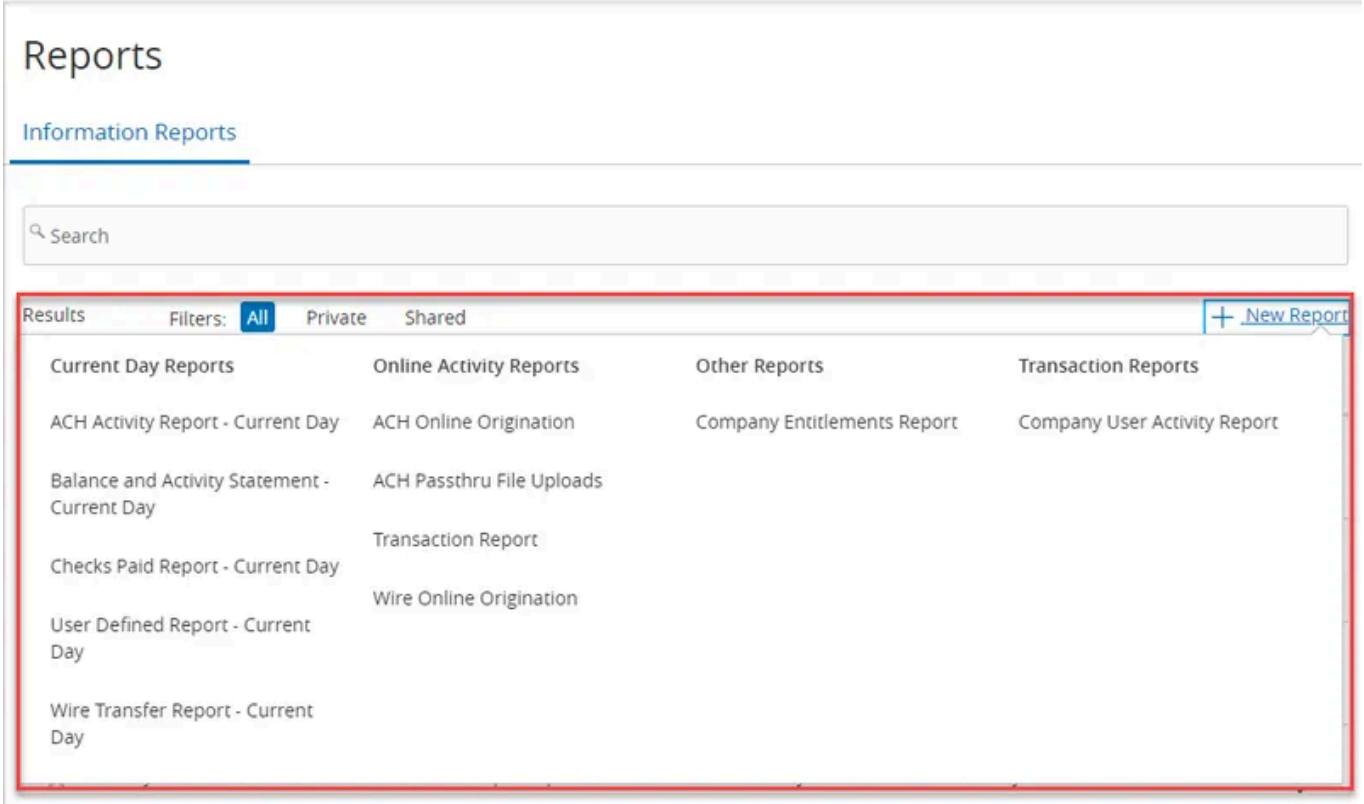
Here you will see two tiles:

Reports - Click on this tile to generate templated/canned reports as well as create brand new reports.

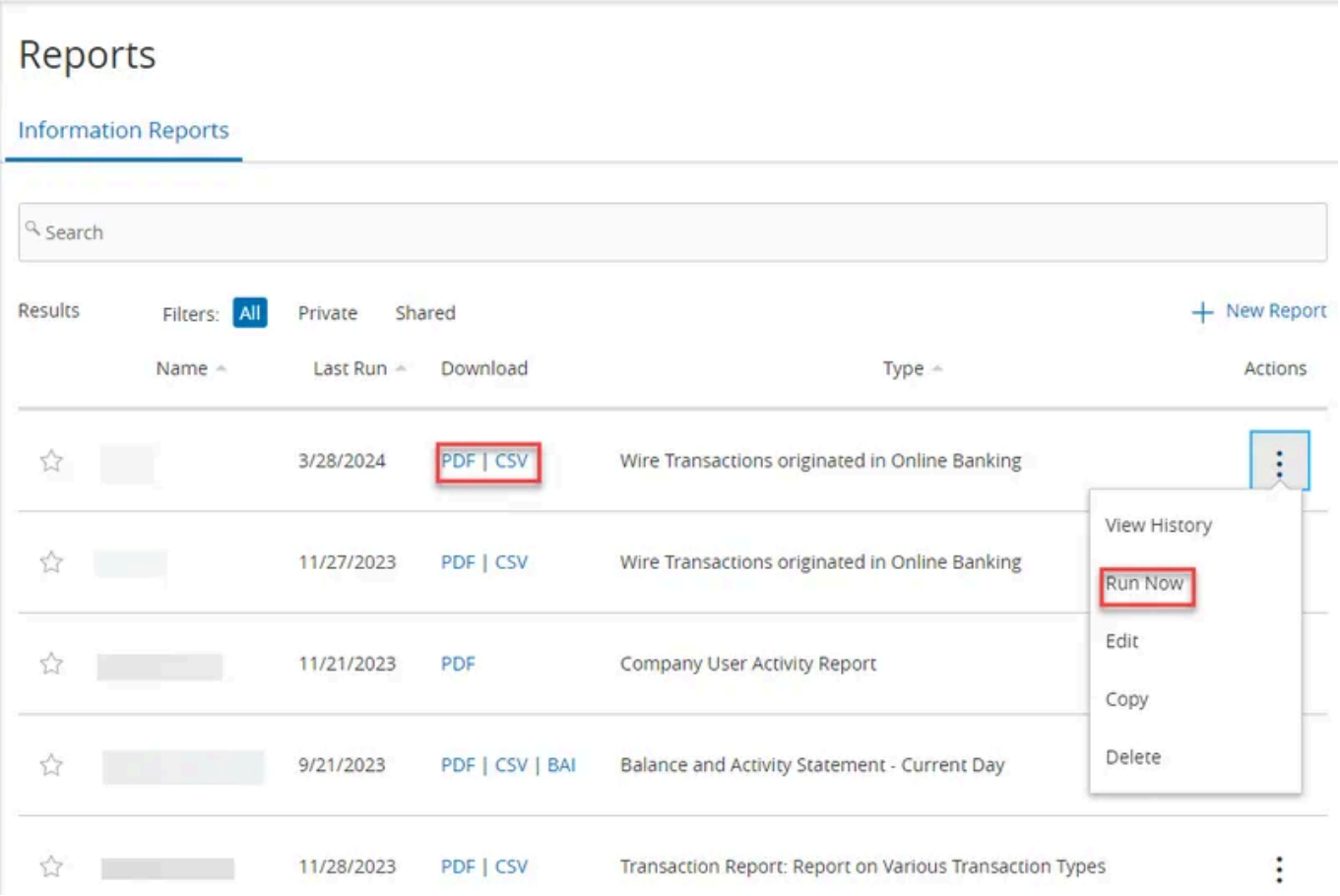
eStatements - Click on this tile to pull monthly statements or to find annual tax documents for accounting purposes.



When you click on the **Reports** tile, you will be brought to the **Information Reports** screen where you can create new reports or pull existing templates.



To generate a report, select the **3-dot menu** next to each existing report template. When the report has generated, the **PDF | CSV** text displays highlighted as a hyperlink. If you click on this link, it will automatically download the file in the format selected.



Reports can be configured to be run on demand, or in a pre-defined pattern.

Reports

Information Reports

Transaction Report

This report will generate the following file formats: PDF, CSV

Do you want this report to be private or shared?

- ○ Private
- ◉ Shared

This report was created by another user and the privacy cannot be changed.

What do you want to name the report?

GoGo Gadget

What account(s) do you want to include?

☐ All Accounts (5)

3 of 5 Accounts selected

What dates do you want to include?

Last Month

How often do you want this report to run?

- ◉ On Demand
- ○ Every Business Day
- ○ Every Calendar Day
- ○ Weekly
- ○ Monthly

Cancel | Save and Run | Save