Guide

Direct Deposit API Guide for ADP Workforce Now

Published on
May 07, 2020 1:53PM

Last modified
Feb 11, 2022 4:09PM

Always Designing for People™

ADP Copyright Information

Published on
May 07, 2020 1:53PM

Last modified
Feb 11, 2022 4:09PM

Table of Contents

Chapter 1
About this API
Summary
What's New in this Guide?
Feb 2022
Supported Product Version and Customer Base
Process Overview
Typical B2B Application Integration Flow
Typical B2C Application Integration Flow
Required Setup Steps
About Direct Deposit
About Prenotification

Chapter 2
API Details
API Endpoints and samples
Application Scope
Supported Actors
Sequence of Interactions
Data Dictionary
Common Exception Responses

Chapter 3
Frequently Asked Questions (FAQs)
Question 1: How do I insert a new Direct Deposit record for an employee?
Question 2: How do I Retrieve the Actual Deposited Amount after each Payroll Run?

Chapter 4
Known Issues and Limitations
US871362: Unable to process the Direct Deposit if the client configured the work flow for the Direct Deposit
Impacted APIs
Issue Description
Suggested Workaround
US1130068: Generate the event notification when changes are made to an employee's Direct Deposit
Impacted APIs

Issue Description
Suggested Workaround

Chapter 5
Appendix

Chapter 1

About this API

Summary

The Direct Deposit Application Programming Interface (API) allows your application to manage an employee's Direct Deposit setup.

What's New in this Guide?

In this section, we will announce any new revisions to the Direct Deposit API and this guide.

Feb 2022

API endpoints and samples are now linked to API explorer.

Supported Product Version and Customer Base

The Direct Deposit API is available for ADP Workforce Now clients in the United States (U.S.) and Canada with Payroll features.

Process Overview

Typical B2B Application Integration Flow

	Actor	Task Description
1	Payroll Practitioner	Setup Direct Deposit accounts in ADP Workforce Now.
2	Your Application	Use the Worker Management API to synchronize employee data between ADP Workforce Now and your application.
3	Payroll Practitioner	In your application, select an employee and initiate an action to view Direct Deposits.
4	Your Application	Use the Direct Deposit API to retrieve Direct Deposit setup data for the selected employee.
5	Payroll Practitioner	In your application, view an employee's Direct Deposit data.
6	Payroll Practitioner	In your application, update an employee's Direct Deposit data.
7	Your Application	Use the Direct Deposit API to replace Direct Deposit setup data for the selected employee.

Result: Upon successful payroll processing, ADP automatically distributes funds from the employee's pay to their Direct Deposit account per the Direct Deposit configuration.

Typical B2C Application Integration Flow

	Actor	Task Description
1	User	Login to your application.
2	User	Authenticate using the employer-issued ADP credential and consent to allow your application to retrieve the individual's ADP-managed Direct Deposit data for that employer.
3	Your Application	Use the Direct Deposit API to retrieve the individual's Direct Deposit setup data.
4	User	In your application, view Direct Deposit setup data.
5	User	In your application, update Direct Deposit setup data.
6	Your Application	Use the Direct Deposit API to replace Direct Deposit setup data for the user under the employer.

Result: Upon successful payroll processing, ADP automatically distributes funds from the employee's pay to their Direct Deposit account per the Direct Deposit configuration.

Required Setup Steps

In ADP Workforce Now, the Direct Deposit is a type of deduction. The Direct Deposit API supports all Direct Deposit codes configured by clients. Clients need to make sure the codes used by your application are added and configured in the ADP Workforce Now user interface (UI) by selecting **Setup > Tools > Validation Tables > Payroll > Deductions and Deposits > Deposits**.

About Direct Deposit

Direct deposit is a banking feature, enabling your employees to have their entire paychecks or a portion of their paychecks deposited automatically into one or more bank accounts. For example, accounts such as savings, checking, money markets, IRAs, and others. A company's setup controls how many Direct Deposits can be configured for an employee. For example, a company's setup might allow employees to add up to three Direct Deposits or up to 500 Direct Deposits.

About Prenotification

Prenotification (or prenote) is when a bank tests new Direct Deposit accounts before actual deposits are made. This is to make sure that the funds are deposited into the correct account. Prenote usually takes 10 days. During prenote, the bank will notify the employer if an invalid routing number or account number was used. The employees will continue to receive a negotiable (live) paycheck during prenote. After prenote, Direct Deposits are made automatically. Due to the prenote process, changes made to a Direct Deposit normally take one or two payroll cycles to be executed.

When an employee's banking information is updated, prenote is restarted, usually for five days. The employee will receive a negotiable paycheck during this time.
In ADP Workforce Now, a client can configure prenote for Direct Deposit. After configured, a user can select to execute the prenote using one of the following options:

- **Company Default** – Use your company's standard prenote end date.
- **No Prenote** – Skip prenote and make the Direct Deposit as soon as possible.
- **Custom** – Select an end date from the calendar.

The prenote option is enabled only for the requested clients. Contact ADP to enable the prenote option for a specific client.

Chapter 2

API Details

API Endpoints and samples

Workforce Now Pay Distribution API

Application Scope

The canonical URI corresponding to the Direct Deposit API needs to be added in the Consumer Application Registry (CAR) for the subscription following which a user can access this API and make successful API calls. The canonical URIs for APIs in this guide:

- **/payroll/payrollManagement/payrollInstructionManagement/payDistributionManagement/worker.payDistribution.read**
- **/payroll/payrollManagement/payrollInstructionManagement/payDistributionManagement/worker.payDistribution.change**

Supported Actors

Request Parameter roleCode Value	**Usage**
Practitioner	Retrieves an employee's Direct Deposits as a practitioner. A system user is considered a practitioner.
Employee	Retrieves one's Direct Deposits as an employee. This is typically used to view one's own Direct Deposits.

Sequence of Interactions

1. Your consumer application makes a request to the ADP API endpoint to GET Pay Distributions for a worker.
2. The ADP API endpoint responds to your consumer application with single worker Pay Distributions information.
3. Your consumer application makes a request to the ADP API endpoint for **/events/payroll/v1/worker.pay-distribution.change** meta.
4. The ADP API endpoint responds to your consumer application with the meta payload, which lists eligible Direct Deposits for the worker under the **/meta/data/transforms/payDistribution/distributionInstructions/depositAccount /financialAccount/typeCode/codeList** collection. Yo

consumer application needs to validate the Direct Deposit Code to be processed is returned in the codeList collection before building the payload for the next call. If the Direct Deposit Code you need to add, update, or delete is not part of the collection, do one of the following: Your application needs to provide user error handling and stop. Go to the next step.

5. Your consumer application makes a request to the ADP API endpoint for the **/events/payroll/v1/worker.pay-distribution.change** API with the payload.
6. The ADP API endpoint responds to the consumer application about the details of the **worker.pay-distribution.change** API.

Data Dictionary

Resources listed in the following table can be accessed in the ADP Workforce Now UI by selecting People > Pay > Pay Profile > Direct Deposits.

Schema Location	Field Name	Is Required (Y/N)	Note
/worker/associateOID	NA	Y	To retrieve the value for an employee, use GET workers.
/payDistribution/itemID	NA	Y	1. To retrieve the item ID for an employee, use **GET/payroll/v2/workers /{aoid}/pay-distributions** API. 2. The item ID differs for the **Particular Company Code** and **Payroll File Number** when an employee has multiple positions.
/effectiveDateTime	**Effective Date**	Y	1. Indicates the date Direct Deposit updates should be effective. 2. **Effective Date** does not apply to the Employee **roleCode**.
/payDistribution/distributionInstructions/instructionStatusCode	**Active**		
/payDistribution/distributionInstructions/distributionPercentage	**Amount to Deposit** Partial Net	Y	Include this **distributionPercentage** field in the request if a percentage of **Pay** should be deposited.
/payDistribution/distributionInstructions/distributionAmount/amountValue	**Amount to Deposit** Percent Net	Y	Include this **distributionAmount** field in the request if fixed amount should be deposited.
/payDistribution/distributionInstructions /depositAccount/fina	**Account Number**	Y	Indicates the Account Number to which the Direct Deposit applies.

ncialAccount /accountNumber			
/payDistribution/distributionInstructions/depositAccount/financialAccount/typeCode/codeValue	The type of direct deposit is displayed.	Y	Indicates the Direct Deposit Code.
/payDistribution/distributionInstructions /depositAccount/financialParty/routingTransitID/idValue	**Routing Number**	Y	(U.S. clients) Indicates the Routing Number to which the Direct Deposit applies.
/payDistribution/distributionInstructions /depositAccount/financialParty /financialPartyID/schemeCode/codeValue	**Bank**	Y	(Canadian clients) Indicates the bank name.
/payDistribution/distributionInstructions /depositAccount/financialParty /branchNameCode/codeValue	**Branch**	Y	(Canadian clients) Indicates the Branch code.
/payDistributions/distributionInstructions/bonusOnlyIndicator	**For Bonus Pay Only**	Y	Indicates the bonus pay for Direct Deposit.
/payDistributions/distributionInstructions/precedenceCode/codeValue	**Priority #**	Y	Indicates priority value for Direct Deposit.
/payDistributions/distributionInstructions/prenote/prenoteOptionCode/codeValue	**PreNote End Date**	Y	Indicates prenote options for Direct Deposit. The Prenote Codes are as follows: • **N** - No Prenote • **O** - Custom • **D** - Company Default
/payDistributions/distributionInstructions/prenote/prenoteDate	**PreNote End Date**	Y	Indicates prenote options date for Direct Deposit.
/payDistribution/distributionInstructions/remainingBalanceIndicator	NA	Y	Indicates the Full Net Direct Deposit. Include **"remainingBalanceIndicator":true** if the Direct Deposit is of Full Net.
/payDistribution/distributionInstructions/itemID	NA	Y	Indicates the **itemID** for a particular deposit code.

Common Exception Responses

You may encounter exceptions outside your common success scenarios. You must account for these exceptions during your initial development.

For more information. see API Common Exceptions and Tips for Handling.

Table below lists common response code and tips to handle for /events/payroll/v1/worker.pay-distribution.change

ResponseCode	Response Condition	message Txt	Tips to Handle
400 Bad Request	Add a Direct Deposit with a routing number in the wrong format.	"**userMessage**": {"**messageTxt**": Routing Number must be 9 digits. Valid values are0 - 9."}	User to check if a valid routing number is passed.
400 Bad Request	Add a Direct Deposit for an existing Deduction Code for an employee.	"**userMessage**": {"**messageTxt**": "This Direct Deposit code was already added as of 12-17-2018. Enter a unique Direct Deposit code."}	User to check if there is an existing deposit of the same Deposit Code.
400 Bad Request	Add a Full Net Direct Deposit when there is an existing Full Net Deposit.	"**userMessage**": {"**messageTxt**": " An employee can have only one full deposit and one bonus deposit."}	User to check if there is an existing full net deposit.
400 Bad Request	Add a Direct Deposit when an invalid **itemID** is passed.	"**userMessage**": {"**messageTxt**": "itemID is invalid."}	User to check if a Valid Item ID is passed.
400 Bad Request	Add a Direct Deposit when the distribution percentage exceeds 100.	"**userMessage**": {"**messageTxt**": "The total for all non-bonus percentage deposits cannot exceed 100 percent."}	User to check that the Percent Net does not exceed 100. **Note:** The total for all non-bonus percentage deposits cannot exceed 100 percent.
400 Bad Request	Add a Direct Deposit when distribution amount is blank.	"**userMessage**": { "**messageTxt**": "Request is invalid." }	User to check whether value is provided for distribution amount. **Note:** Values should be provided for all the mandatory fields.
400 Bad Request	Add a Direct Deposit when invalid financial Account/typeCode is passed in their request.	"**userMessage**": {"**messageTxt**": "financialAccount/typeCode is invalid."}	Your application should validate the code using values returned in the meta API.
400 Bad Request	Update the Direct Deposit with invalid Item ID.	"**userMessage**": {"**messageTxt**":"instruction/itemID is invalid."}	User to check a valid item ID associated with the deposit code is passed when updating the Direct Deposit.
400 Bad Request	Update the Direct Deposit with invalid Routing number.	"**userMessage**": {"**messageTxt**": "The check digit for the Routing number is invalid."}	User to check that a valid routing number is passed.
400 Bad Request	Update the Direct Deposit with invalid financial **Account/typeCode**.	"**userMessage**": {"**messageTxt**": "The Deposit could not be found."}	Your application should validate the code using values returned in the meta API.
400 Bad Request	Add a Direct Deposit prior to worker hire date (future hired).	"**userMessage**":{"**messageTxt**": "Select an effective date of 07-01-2019 or later. That's when the employee's first Position/record took effect."}	
400 Bad Request	A user has an existing bonus pay **fullnet** Direct Deposit. Add another Full Net Direct Deposit with a different code and bonus pay.	"**userMessage**": {"**messageTxt**": "An employee can have only one bonus deposit."}	

400 Bad Request	A user has an existing bonus pay **fullnet** Direct Deposit. Add another full net Direct Deposit with the same code and bonus pay.	"**userMessage**": {"**messageTxt**": "This Direct Deposit code was already added as of 03-26-2019. Enter a unique Direct Deposit code."}	
400 Bad Request	A user has an existing bonus pay partial **net** Direct Deposit. Add another partial net Direct Deposit with a different code and bonus pay.	"**userMessage**": {"**messageTxt**": "An employee can have only one bonus deposit."}	
400 Bad Request	A user has an existing bonus pay partial **net** Direct Deposit. Add another partial net Direct Deposit with the same code and bonus pay.	"**userMessage**": {"**messageTxt**": "This Direct Deposit code was already added as of 03-26-2019. Enter a unique Direct Deposit code."}	
400 Bad Request	Add a Direct Deposit with custom prenote and without date.	"**message**": "An error has occurred. Please contact your ADP Representative for further information."	
400 Bad Request	Add a Direct Deposit with the same existing priority number.	"**userMessage**": {"**messageTxt**": "A deposit was already added with this priority number. An employee cannot have more than one deposit with the same priority number."}	
400 Bad Request	Add Direct Deposit code which is not in the validation table.	"**userMessage**": {"**messageTxt**":"financialAccount /typeCode isinvalid."}	Make your end user aware that the code is not valid.

Chapter 3

Frequently Asked Questions (FAQs)

Question 1: How do I insert a new Direct Deposit record for an employee?

Answer: For more information, see the Tip in Chapter 2 - Retrieving an Employee's Direct Deposit under the API Usage section.

Question 2: How do I Retrieve the Actual Deposited Amount after each Payroll Run?

Answer: The Direct Deposit API allows applications to view and update Direct Deposit setups only. To retrieve the actual deposited amount for each account after each payroll run, your application needs to use either the Pay Statement API or Pay Data Output API.

Chapter 4

Known Issues and Limitations

US871362: Unable to process the Direct Deposit if the client configured the work flow for the Direct Deposit

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v1/worker.pay-distribution.change	Practitioner

Issue Description

When trying to update the Direct Deposit record using the impacted API, the response does not indicate the request is going through the workflow.

If your application tries to retrieve the employee's Direct Deposit, prior data will be returned instead of the changes submitted.

Suggested Workaround

There are no workarounds available.

US1130068: Generate the event notification when changes are made to an employee's Direct Deposit

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v1/worker.pay-distribution.change	Practitioner

Issue Description

Currently, there are no supported event notifications for the Direct Deposit API.

Suggested Workaround

As employees and client practitioners could use ADP Workforce Now to update Direct Deposits; if your application stores the employees' Direct Deposit setup then it needs to build a regular data retrieval process to keep your application data in synchronizations with ADP Workforce Now.

Chapter 5

Appendix

The Job Requisition API supports an approval workflow. The following steps need to be completed:

1. Log in as a Standard Practitioner in ADP Workforce Now.
2. In the ADP Workforce Now UI, navigate to **SETUP > Approval Process > Activity Configuration** and select the **Standard Activities** tab.
3. From the drop-down list under **Role**, select **Practitioner**.
4. Find the job requisition.
5. For the job requisition, enable the **Approval Path** (slide the indicator to GREEN).
6. Click **Edit**.
7. Click **Approval Path**. From the drop-down list, select **ADD APPROVAL PATH**.
8. Provide the path name. Then, next to the job requisition, add the description.
9. From the drop-down list next to **WHERE DOES THIS GO?**, select the appropriate value (for example, Specific User).
10. From the drop-down list next to **ASSIGNED TO**, select the user or type the user's name.
11. Next to **STEP TYPE**, select **Approval**.
12. Click **Save**.

Tip

In the case you have set the **APPROVAL PATH**, by default, all the requisitions created by the Hiring Manager will go for approvals, and the status of the job requisition will be **PENDING APPROVAL**.

If you don't want all the requisitions created by the Hiring Manager to go for approvals then, while creating the approval path, you need to define **Approvals Not Required For**. Approvals are bypassed, and employee information updated immediately if a member of this group submits the activity.