# Return From Leave Workflow

**Step 1.**

**Identify Returned Employees**

From FIN284 report, business is responsible to clean and produce a list of employees where the following conditions are met:

* Not future dated i.e. paycheck issue date is not greater than today’s date
* Payroll status neither: terminated, retired, or deceased
* Elig Fld 1 flag neither ‘G’ or ‘N’
* Pay Status is not ‘Paid Leave’
* Has no completed or pending recon status based on RFL Master File that business kept for logging and tracking purpose.
* No duplicate EEID

Once FIN284 report is cleaned, the only input the bot needs is an input Excel workbook consisting of 1 worksheet with following columns:

* Business Unit
* EEID – Employee ID
* Payroll Status
* Pay Status
* Pay Group
* Current Date
* Action Date
* Elig Fld1
* Elig Fld2
* Empl Record
* Job Indicator

Business is responsible for dropping the **input file** in RPA Box folder (path: Box\RPA\{ENV}\Return From Leave Reconciliation\{ENV} File Drop)

**Step 2.**

**AA Bot Initiation**

Below are initiation steps performed by AA bot:

1. Setup Runtime Directories
   1. Ensure required folders (e.g., Logs, Input, Output, Exceptions, Archive, Config, Process Logs, Snapshots) exist.
   2. Create them if missing and apply appropriate access controls (ACLs).
2. Load Configuration
   1. Read the bot configuration file (Config.json or .xml) to retrieve paths, credential keys, logging level, and email distribution lists.
3. Clean Up Residual Applications
   1. Terminate or close leftover instances of applications like Excel, PeopleSoft browser sessions, and Box drive.
4. Initialize Logging
   1. Set up Action and Exception logs with headers.
   2. Generate a unique run ID (UUID) and log the “Bot Start” event

**Step 3.**

**AA Bot Downloading Input File into EC2 and Reading Into Main Data Table.**

Inside EC2 server, Bot will download the input file dropped in RPA Box folder (path:Box\RPA\{ENV}\Return From Leave Reconciliation\{ENV} File Drop) from AA control room utilizing AA Box package (API). Credentials required to establish authentication are stored within config.json inside AA control room – RFL folder and credential stored in AA control room locker.

The downloaded input file (Excel) will be temporarily stored in EC2 directory: C:/ProgramData/AutomationAnywhere/Bots/Logs/RFL-UCPath/Input

The purpose of downloading the input file into EC2 directory is for bot to read the data utilizing AA Excel Basic action and stores into data table (or refer to as main data table).

**Step 4.**

**Identify Returned Employees by Payroll Status, Job Indicator, and Arrear Balance. Ready for Reconciliation.**

*Job Data check*

Bot initiates a login to Peoplesoft and navigate to Job Data page. Once login is established and arrive at Job Data page. Bot iterates the list of EEID from main data table and as it lands on Job Data page, it reads and captures the following data points:

* Payroll Status – in Work Location tab
* Job Indicator – in Work Location tab
* Pay Group – in Payroll tab (NOTE: pay group is to indicate whether EE is monthly or biweekly employee)

*Rules for Exception*

As a process of determining reconciliation, any EEID with the following conditions will be removed from main data table and place in exception data table:

* Payroll Status is not Active
* Job Indicator is not Primary Job

All EEID with above conditions are placed in exception data table with exception status :”Payroll Status or Job Indicator did not meet criteria”

*Arrear Balance check*

Bot then navigates to Arrears page, and as bot iterates the list of remaining EEID in main data table, bot checks for UCS arrears records.

*Rules for Exception*

As a process of determining reconciliation, any EEID with arrear balance > $0 will be removed from main data table and place in exception data table. Exception status should be: “arrears detected”.

The main data table now consists of recon-eligible EEID.

**Step 3.**

**Review Payroll Deductions**

Bot then navigates to Query Viewer page; search for FIN001 query viewer. As bot iterates the list of remaining EEID in main data table, bot downloads FIN001 report (in .xls) into EC2 repository: C:/ProgramData/AutomationAnywhere/Bots/Logs/RFL-UCPath/Input/FIN001

*Payroll Monthly and Biweekly Calendars*

Payroll Monthly and Biweekly calendars are essential sources for identifying possible missed paychecks within FIN001 reports. These calendars are hosted in the following Box folder : Box\#UCPC-FN-BB\#PII-UCPC-FN-BB\Return From Leave\RPA Recon\Resource Documents

For one time only, bot will download the payroll calendar hosted above from AA control room utilizing AA Box package (API). The calendar will be stored inside UCPATH > Utilities folder within AA control room. Credentials required to establish authentication are stored within config.json inside AA control room – RFL folder and credential stored in AA control room locker.

*Dataframe Initiation, Missing Dates Identification, Inserting Missed Paychecks*

Once bot finishes downloading FIN001 for all EEID, bot triggers Python script to perform the following data processing:

1. Read payroll calendar into payroll dataframe
2. Read FIN001 sheet into FIN001 dataframe
3. Identify missing paycheck issue dates by comparing dates on FIN001 and payroll dataframe per selected plan types below:
   1. Legal Insurance
   2. Supplemental Life
   3. Employee & Dependent AD&D
   4. Basic Dependent Life
   5. Exp Dependent Life - Spouse/DP
   6. Exp Dependent Life - Child
   7. Voluntary Long-Term Disability
   8. Voluntary Short-Term Disability
   9. Accident
   10. Critical Illness EE
   11. Critical Illness SP
   12. Hospital Indemnity
   13. Dental \*
   14. Vision \*
   15. Life \*
   16. Basic Disability \*

\* for bi-weekly these plan types are deducted in monthly basis

1. Inserting missed paychecks as identified by missing paycheck issue dates (step 3 above) given the selected plan types above. Highlight the inserted missed paycheck by setting cell’s color.
2. Identify variances in current deduction for each selected plan types above in month-to-month basis and highlight by changing cell’s color.

*Rules for Exception*

If at any time either of following condition is met, the script processing of FIN001 report should stop:

1. If more than 1 pay groups identified. Exception status should be: “multiple pay groups identified”.
2. If there is no missed paychecks on selected plan types identified. Exception status should be:”no missed paychecks”

Bot skips to the next EEID and log the EEID to exception table.

At this point, FIN001 dataframe should consist of identified missed paychecks, variances for selected plan types.

**Step 4.**

**Oracle Transaction Register**

Bot initiates a login to Oracle Cloud Financial (url: https://elda.fa.us2.oraclecloud.com/) and navigate to Page 1 page. Path: Oracle Cloud Financial > Home > Tools > Reports and Analytics > page 1 . In page 1 page, bot inserts the EEID prefixed with ‘e’, for example: e10012345, into Bill-to-Customer Account field and extract the transaction register analysis report (in csv). Note: Steps are described in Current Process section above

The transaction register analysis report will be processed under Python script as follows:

1. Read transaction register analysis report into transaction register dataframe
2. Format transaction register dataframe following business rules.
3. Append transaction register dataframe into the cluster of dataframes that include FIN001 dataframe

A screenshot of a computer

AI-generated content may be incorrect.

**Step 5.**

**Export Payments from Oracle**

Bot then navigates to Oracle Financials > Home Screen > Receivables > Billing > Review Customer Account Details. Bot inserts the EEID prefixed with ‘e’, for example: e10012345, into Customer Number field and extract the payment report. Note: Steps are described in Current Process section above

The payment report will be processed under Python script as follows:

1. Read payment report into payment dataframe
2. Format payment dataframe following business rules.
3. Append payment dataframe into the cluster of dataframes that include FIN001 dataframe

A screenshot of a computer

AI-generated content may be incorrect.

**Step 6.**

**Data Processing of Payroll Deduction (FIN001) Report**

At this step we have the following dataframes:

1. FIN001
2. Transaction Register
3. Payment

Finally, Python script will perform a pivot table conversion on FIN001 dataframe and instantiate Summary dataframe.

All dataframes will then be compiled into individual worksheet under the final Excel report.

1. FIN001 – will be FIN001 worksheet under final Excel report
2. Transaction Register – will be Transaction Register worksheet under final Excel report
3. Payment - will be payment worksheet under final Excel report
4. Summary - will be Summary worksheet under final Excel report

**Step 7.**

**Output**

Upon completion of bot run, bot is going to perform the following:

* Migrate logs, final reports (Excel), input file, and artifacts from EC2 directory to the respective RPA Box folders utilizing Box API.
* Send email confirmation to business with summary of the run.