

# invoice-processing

Type: Combined Analysis (Multiple Files) Generated: 2026-01-29 12:15:41 Processing Time: 37701ms

---

```
# Business Logic for Invoice Processing
```

This document outlines the business rules governing the invoice processing workflow, based on an analysis of the RPG programs `FO605R` and `FO709R`. The primary focus is on the conditions and operations involved in preparing and processing invoices and credit notes, as well as managing order statuses.

The core logic for invoice processing is contained within the `FO605R` program, which handles the preparation and printing of invoices/credit notes, and the `FO709R` program, which updates order header information to reflect the current processing state. These programs interact to ensure that invoices are processed correctly and that order statuses are updated consistently.

## ## Order Status and Header Rules

1. \*\*Order Already Being Processed (`FO605R`, `FO709R`)\*\*
  - \* \*\*Logic:\*\* Both programs ensure that an order is not processed if it is already being handled by another process.
    - \* \*\*File:\*\* `FOHELP` (Order Header File)
    - \* \*\*Field:\*\* `FOKOUS`, `FOKOWS`
    - \* \*\*Condition:\*\*
      - In `FO709R`, the `FOKOUS` (user) and `FOKOWS` (workstation) fields are updated to reflect the current user and workstation processing the order. This prevents simultaneous processing by multiple users.
      - In `FO605R`, the `FA730R` subprogram is called to check if invoice processing (`w\_kode = 01`) or accounting transfer (`w\_kode = 02`) is already active. If either process is active (`w\_onof = 1`), the program exits.
2. \*\*Valid Period Check (`FO605R`)\*\*
  - \* \*\*Logic:\*\* Ensures that the invoice period is valid and within acceptable limits.
    - \* \*\*File:\*\* `RAAIL1R` (Status File)
    - \* \*\*Field:\*\* `RA1PER`, `RA1AAR`
    - \* \*\*Condition:\*\*
      - The invoice period (`f2peri`) must not be blank (`f2peri = 0`).
      - The period must be a valid date (`test(d)`).
      - The year must not deviate by more than one year from the current year (`w\_avik < -1 OR w\_avik > 1`).
      - The period must not be earlier than the "closed period" (`w\_peri <= w\_sp\_per`).
3. \*\*Valid Invoice Date Check (`FO605R`)\*\*
  - \* \*\*Logic:\*\* Ensures that the invoice date is valid and falls within the specified period.
    - \* \*\*File:\*\* None (local validation)
    - \* \*\*Field:\*\* `F2DATO`
    - \* \*\*Condition:\*\*
      - The invoice date (`f2dato`) must be a valid date (`test(d)`).
      - The month and year of the invoice date must match the period (`w\_mnd1 = w\_mnd2` and `w\_aar1 = w\_aar2`).
4. \*\*Order Type Restriction (`FO605R`)\*\*
  - \* \*\*Logic:\*\* Only processes invoices and credit notes (order type 3).
    - \* \*\*File:\*\* None (local validation)
    - \* \*\*Field:\*\* `VAOAKK`
    - \* \*\*Condition:\*\* The order type (`vaoakk`) must be equal to `3`.

```

5. **Order Header Update (`FO709R`)*
   * **Logic:** Updates the order header with the current user, workstation, and timestamp information.
     * **File:** `FOHELP` (Order Header File)
     * **Field:** `FOKOUS`, `FOKOWS`, `FOKODA`, `FOKOTI`, `FOEDAT`, `FOETIM`, `FOEUSR`
     * **Condition:**
       - The `FOKOUS` and `FOKOWS` fields are updated with the current user (`l_user`) and workstation (`l_wsid`).
       - The `FOKODA` and `FOKOTI` fields are updated with the current date and time.
       - The `FOEDAT` and `FOETIM` fields are updated with the timestamp of the update.
       - The `FOEUSR` field is updated with the current user.

```

#### ## Subprogram Calls Affecting Logic

Beyond direct file checks, several external subprograms are called that play a significant role in the workflow:

1. \*\*`FA730R` (Check Active Processes)\*\*
  - \* \*\*Trigger:\*\* Called in `FO605R` before processing begins.
  - \* \*\*Logic:\*\* Checks whether invoice processing (`w\_kode = 01`) or accounting transfer (`w\_kode = 02`) is already active.
  - \* \*\*Impact:\*\* If either process is active (`w\_onof = 1`), the program exits to prevent conflicts.
2. \*\*`FA720R` (Update Job Register)\*\*
  - \* \*\*Trigger:\*\* Called in `FO605R` after validation checks are complete.
  - \* \*\*Logic:\*\* Updates the job register to indicate that invoice processing is active (`w\_onof = 1`).
  - \* \*\*Impact:\*\* Prevents other processes from starting while the current process is active.
3. \*\*`FA920R` (Generate Member Name)\*\*
  - \* \*\*Trigger:\*\* Called in `FO605R` during initialization.
  - \* \*\*Logic:\*\* Generates a unique member name for the output file.
  - \* \*\*Impact:\*\* Ensures that each output file has a unique identifier.
4. \*\*`FO604C` (Check Member Usage)\*\*
  - \* \*\*Trigger:\*\* Called in `FO605R` during initialization.
  - \* \*\*Logic:\*\* Checks whether the generated member name has already been used.
  - \* \*\*Impact:\*\* Prevents duplicate member names.
5. \*\*`FO612C` (Start Printing)\*\*
  - \* \*\*Trigger:\*\* Called in `FO605R` after all validation checks are complete.
  - \* \*\*Logic:\*\* Initiates the printing of invoices or credit notes.
  - \* \*\*Impact:\*\* Starts the actual output process for the selected orders.

#### ## Differences Between Programs

1. \*\*Primary Focus:\*\*
  - `FO605R` focuses on preparing and printing invoices/credit notes, including validation of periods and dates.
  - `FO709R` focuses on updating the order header to reflect the current processing state.
2. \*\*File Usage:\*\*
  - `FO605R` interacts with multiple files (`RAA1L1R`, `FPM1LUR`, etc.) to validate periods, generate member names, and update job registers.
  - `FO709R` primarily interacts with the `FOHELP` file to update order header information.
3. \*\*Subprogram Calls:\*\*
  - `FO605R` calls several subprograms (`FA730R`, `FA720R`, `FA920R`, `FO604C`, `FO612C`) to handle validation, job registration, and printing.
  - `FO709R` does not call any external subprograms, focusing solely on updating the order

header.

#### ## Summary

The `FO605R` and `FO709R` programs work together to ensure accurate and conflict-free invoice processing. `FO605R` handles the preparation, validation, and printing of invoices/credit notes, while `FO709R` updates the order header to reflect the current processing state. Together, they ensure that invoices are processed efficiently and that order statuses are updated consistently to prevent conflicts and maintain data integrity.