

RA107R.MBR

Type: Single File Analysis Path: NXCLOUD/rpgsrc/RA107R.MBR Generated: 2026-01-29 12:15:41

Processing Time: 20479ms

```
# Business Logic for RA107R
```

This document outlines the business rules that govern the maintenance of department records, based on an analysis of the RPG program `RA107R`. The primary focus is on the conditions that will **block** an operation from being processed.

The core logic for maintaining department records is contained within the `xc2bld` subroutine in `RA107R`. A department record must pass all the following checks to be processed.

```
## Order Status and Header Rules
```

1. **Department Code Validation**

* **Logic:** The department code (`C1GKOD`) must not be zero. If it is zero, the operation is blocked, and the user is prompted to enter a valid department code.
* **File:** `RA07LR` (Department Master File)
* **Field:** `C1GKOD`
* **Condition:** The process will not proceed if `C1GKOD = 0`. This ensures that a valid department code is provided.

2. **Duplicate Department Check**

* **Logic:** The system checks if the department code already exists in the `RA07LR` file. If it does, the user is prompted with a message, and the operation is blocked unless the user chooses to overwrite or cancel.
* **File:** `RA07LR` (Department Master File)
* **Field:** `RA07LR_GKOD`
* **Condition:** The process will not proceed if `CHAIN RA07LR_KEY` finds an existing record with the same `RA07LR_GKOD`. This prevents duplicate department codes.

```
## Configuration and Authorization Rules
```

3. **Postal Code Validation**

* **Logic:** If a postal code (`C2GPNR`) is provided, the system validates it by calling the external program `RS730R`. If the postal code is invalid, the operation is blocked, and the user is prompted to correct it.
* **File:** `RA07LR` (Department Master File)
* **Field:** `C2GPNR`
* **Condition:** The process will not proceed if `P_RET` returned from `RS730R` is not `BLANK`. This ensures that only valid postal codes are accepted.

4. **Delivery Type Validation**

* **Logic:** If a delivery type (`C2GLTY`) is provided, the system validates it by calling the external program `RY500C`. If the delivery type is invalid, the operation is blocked, and the user is prompted to correct it.
* **File:** `RA07LR` (Department Master File)
* **Field:** `C2GLTY`
* **Condition:** The process will not proceed if the delivery type is not valid as determined by `RY500C`. This ensures that only valid delivery types are accepted.

```
## Financial and Transactional Rules
```

5. **Field Validation for Mandatory Fields**

* **Logic:** The department description (`C2GTXT`) must not be blank. If it is blank, the

```

operation is blocked, and the user is prompted to enter a valid description.
*   **File:** `RA07LR` (Department Master File)
*   **Field:** `C2GTXT`
*   **Condition:** The process will not proceed if `C2GTXT = *BLANK`. This ensures that every
department has a meaningful description.

## Special Conditions (Program-Specific)

6. **Subfile Record Selection**
*   **Logic:** When navigating or updating subfile records, the system ensures that only
records matching the current firm (`W_FIRM`) are processed. If a record does not match the firm,
it is skipped.
*   **File:** `RA07L1`, `RA07L2` (Subfile Data Files)
*   **Field:** `RAGFIR`
*   **Condition:** The process will not include records where `RAGFIR <> W_FIRM`. This ensures
that only records relevant to the current firm are displayed or modified.

7. **Deletion Validation**
*   **Logic:** Before deleting a record, the system verifies that the record exists in the
`RA07LR` file. If the record does not exist, the deletion is blocked.
*   **File:** `RA07LR` (Department Master File)
*   **Field:** `RA07LR_GKOD`
*   **Condition:** The process will not delete a record if `CHAIN RA07LR_KEY` does not find a
matching record. This prevents accidental deletion of non-existent records.

## Subprogram Calls Affecting Logic

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

1. **`RS730R` (Postal Code Validation)**
*   **Trigger:** Called when a postal code (`C2GPNR`) is entered or modified.
*   **Logic:** Validates the postal code and returns the corresponding city name (`P_STED`) or
an error code (`P_RETAK`).
*   **Impact:** Blocks the process if the postal code is invalid (i.e., `P_RETAK <> *BLANK`).

2. **`RY500C` (Delivery Type Validation)**
*   **Trigger:** Called when a delivery type (`C2GLTY`) is entered or modified.
*   **Logic:** Validates the delivery type and returns a description (`P_LTXT`).
*   **Impact:** Blocks the process if the delivery type is invalid.

3. **`AF010R` (Postal Code Lookup)**
*   **Trigger:** Called when the user requests a lookup for a postal code.
*   **Logic:** Retrieves the postal code and corresponding city name.
*   **Impact:** Facilitates user input but does not directly block the process.

4. **`XC1MSG` (Duplicate Record Message)**
*   **Trigger:** Called when attempting to create a record with a duplicate department code.
*   **Logic:** Displays a message prompting the user to overwrite or cancel.
*   **Impact:** Blocks the process unless the user chooses to overwrite.

5. **`XD1WIN` (Delete Confirmation)**
*   **Trigger:** Called when a delete operation is initiated.
*   **Logic:** Displays a confirmation window for the delete action.
*   **Impact:** Ensures that deletions are intentional and prevents accidental data loss.

6. **`XC2BLD` (Record Maintenance)**
*   **Trigger:** Called during the creation, update, or view of a department record.
*   **Logic:** Handles the main logic for maintaining department records, including validation
and updates.
*   **Impact:** Central to the program's functionality; blocks invalid operations.

```