# DEV TICKET N° 123456

***Technical file***

| | |
|---|---|
| **Date :** | **[2018]** |
| **Patch version :** | 123 |
| **Author :** | Wafaa Namasse |

## 1. Brief problem description

In the Rate Management window, the « Update rates per center » menu allows to create rates for a given center. After selecting a center, adding a rate requires to enter the stay rate code, then a search is made to find it and display its label. Currently, this search returns an « Unknown rate code » error in some cases, while the rate is located in the correct entity. This error isn't systematic.

## 2. Analysis

This error stems from the fact the rate search doesn't take into account the selected entity, but, in some cases, the default entity. The search process must therefore be modified, so that the entity is systematically filled in.

The concerned grid in the RATEMGT project is *rinput.cpp*. The field validation function that interests us (rate code) is *validate_rinput_ratcod*. Inside, there's the call to the *A10001* API : it is the one that needs modifying.

This API uses data from the *i13* property segment : the *i13.h* header needs therefore to be edited to add a parameter, the entity.

Once this is done, this new parameter will be used, if not empty, in the SQL query built in the *A10001.PC* API, through the *build_search_stay_rates* function, for both R1 and R2 variables. So instead of using the default entity in some cases, the entity selected by the user will systematically be sent in the query, via the *i13* segment.

Then, the API call on the front end must use the entity parameter as well. The signature of the *call_A10001* method will be modified, both in the *cpp* class and its header. In this same file, the entity will also be added as class attribute, so it can fill in the new segment parameter, and be sent to the API query.

Finally, in the *rinput.cpp* grid, the entity field value will be kept in a parameter, that will be sent to the modified *call_A10001* method. The entity is displayed in a dropdown list in case of rate creation, and in the label in case of rate modification (it will then be extracted from the *fcteditctr* list, where the label is from).

# 3. Detailed solution

**General principle** : Adding the entity as a parameter of the rate search, both in the *A10001* API and in the front end.

The following changes have been made :

## 1 - Evolution of the i13 segment :

* In the **\...\i13.h** header, I have added :

```
#define  i13_entity       15

#define  n13_nbParam      15
```

## 2 - Evolution of the A10001 API :

* In the **\...\a10001.PC** file, and the *build_search_stay_rates* function :

- Line 207, I deleted the current code, that fills R1's entity with the default entity, and replaced it with :

```
if(Query().Segment("i13",1).ParameterAt(i13_entity, FALSE).isNull() == FALSE)
{
      if (add == 1 ) query += query_rest;
      query += CParam(" R1.entity = ") + rating +
                               (CParam)Query().Segment("i13",1).ParameterAt(i13_entity,
FALSE) + rating;
       add = 1;
 }
```

- Likewise, line 383, I deleted the R2 entity processing and set in its place the entity selected by the user, which has been saved as R1's entity :

```
       query += (CParam)("R2.entity = R1.entity  AND ");
```

## 3 - Dialog-side evolution:

* In the **\...\ac10001.h** header, I changed the following method’s signature to call with the entity :

```
CString call_10001(int code, CString cod_tar, CString ent_num);
```

* In the **\...\ac10001.cpp** source file, I changed the following method’s signature and added the entity as well for the call :

```
CString DIALOG::call_A10001(int code,CString cod_tar, CString ent_num)
{
while (a10001.FunctionalCode() == "A3")  //    There are still responses
    {
      //...
        A10001.Query().Segment("i13",1).ChangeParameter(i13_entity, entity)  ;
      //…
      }
}
```

* In the **\...\rinput.cpp** file, in the *validate_rinput_ratcod* function, I retrieved the value from either the entity field or the *fcteditctr* list depending on the case :

```
CString s_entity = "";
if(type_operation == "modification")
{
      s_entity = fcteditctr_list("entity");
}
else // rate creation
{
      s_entity = FIELD("libentity");
}
```

- And changed the following line in order to send the entity in the API call :

```
CString label = call_A10001(SCH, s_rinput_ratcod, s_entity);
```

## 4  - UI correction :

* In the Panther UI software, I opened the **\...\rinput.jam** grid

- And corrected the « Budget Letter » label that had a typo *[screenshot before]* → *[screenshot after]*

# 4. Tests

## Tested dialogs

- **Administrative :**
  **1) Rate Management**

### *1) Rate Management dialog test*

**<u>Previous behavior</u>**

This test requires activating Multi-Entity mode (administrative filter).

In the Rate Management dialog, and more precisely in the « Rates → Update rates per center » menu, it's possible to search rates by entering a value in the « Stay rate code » field. For example, the RC1 code below, which exists in the HOS entity, is successfully found when selecting HOS. The problem mentioned in 1. is not reproduced.

*[screenshot of the previous version showing the Rate Management dialog and « Update rates per center » screen, with the RC1 code successfully found with the matching label]*

Previously, the SET code wasn't found in the 058 entity :

*[screenshot of the previous version showing the SET code triggering the « Unknown rate code » error pop-up window]*

However, the SET code does exist for this entity in the database :

*[screenshot of the Oracle SQL query and output, showing one row of results for entity 058 and code SET]*

**<u>Post-correction behavior</u>**

For this test, we're back to the Rate Management dialog's start screen, then the « Rates → Update rates per center » screen.

*[screenshot showing how to select the « Update rates per center » menu within the Rate Management dialog]*

Then any center can be selected, followed by the « Add » button.

*[screenshot showing the « Update rates per center » screen with a center highlighted, as well as the « Add » button]*

The « Rate creation for a center » (*rinput*) grid is open.

After entering « SET » in the « Stay rate code » field, the program does find the matching rate this time, for the 058 entity.

*[screenshot of the « Rate creation for a center » window, with « SET » typed/highlighted in the « stay rate code » field, and the matching label displayed next to it (« SET test label »)]*

Likewise, the RC1 rate, present in the HOS entity, is found successfully.

*[screenshot of the « Rate creation for a center » window, with « RC1 » typed/highlighted in the « stay rate code » field, and the matching label displayed next to it (« RC1 rate »). The HOS entity is shown as the current value of the entity dropdown list, and is highlighted as well]*

However, the « SET » rate is not found in the HOS entity, which is expected behavior since « SET » has been defined only for the 058 entity.

*[screenshot of the « Rate creation for a center » window, with « SET » and « HOS » respectively highlighted in the « stay rate code » and « entity » fields. A pop-up error displays : « Unknown rate code »]*

Rate code search does therefore work for a rate's entity, and only this one.

These tests have been made while creating rates (« Add » button), but the implemented solution remains valid for rate modification, after similar tests passed.

# 5. Impacted files

| Server-side | Dialog |
|---|---|
| - `\...\i13.h`<br>- `\...\a10001.PC` | - `\...\ac10001.cpp`<br>- `\...\ac10001.cpp`<br>- `\...\rinput.cpp` |
| **Delivered components** | |
| - `\...\ratemgt.exe`<br>- `\...\rinput.lib (front end)` | |

# 6. Functional Guide documentation

## 6.1. Rate code search per entity

For a user having a Multi-Entity configuration, the rate search per center in the Rate Management dialog (« Update rates per center » menu) was incorrect : at entering a rate code, the search was launched without taking the entity in consideration, and threw an error while the right entity for this rate had been selected.

A fix has been made so that the search returns the correct results, by systematically taking the entity into account.

This fix closes the following bug ticket :

- 123456

*Ref : 123456*