

# 今からでも遅くない! RPG IIIユーザーに贈る RPG IVの魅力と可能性

## 第2回 驚くほど簡単になる、RPG IVの日付計算機能

**中村 潤** 株式会社アイ・ラーニング  
IT 研修本部 IBM 製品研修部  
ラーニング・アドバイザー

前回は、RPG IVとRPG IIIを比較し、関数の使用とフリーフォーマットについて解説しました。RPG IIIにはない機能が使用でき、より見やすいコーディングが可能になることをおわかりいただけたかと思います。今回は、RPG IVにおける日付計算命令についてご紹介します。これもRPG IIIをお使いの方には、きっと驚きの機能とっていいでしょう。

### RPG IIIの日付計算

最初に、RPG IIIで日付計算を行う場合のロジックを考えてみましょう。

IBM iのデータベースでは日付形式と時間形式のフィールドを定義できますが、RPG IIIではそれらのフィールドを扱える機能がないので、日付は8バイト、時間は6バ

イトの10進数フィールドで定義することが多いと思います。

**図表1**のように入力仕様書でデータ構造を使い、日付8バイト中、1~4バイトを年、5~6バイトを月、7~8バイトを日に分けて、数字フィールドとして計算に使用します。

10日後の日付を算出したい場合、日のフィールド(HI)に10を足し算しますが、結果が月末の日付よりも大きくなった場合はHIから月末の日付を差し引き、月のフィールド(TSUKI)に1を足します。これは月によって月末の日付が違うので、やり方としては泥臭いですが、IF命令やSELECT命令を使って、月末の日付を判断するしかありません。また2月の場合も4年に1度の閏年をどう判断するか、これもやり方がいろいろあります。4で割って余りがなければ閏年とする、とか、あるいはプログラムの寿命を20年と考えて、16年、20年、24年、28年、32年

図表1 RPG IIIでの日付計算

```
FMT I      .....IFILENAMESQNORIP0S1NCCPOS2NCCPOS3NCC.....
0003.00    I              DS
FMT J      .....I.....PFROMTO++DFIELD+L1M1FRPLMNZR.
0004.00    I              1      80HIZUKE
0005.00    I              1      40NEN
0006.00    I              5      60TSUKI
0007.00    I              7      80HI
```

# RPG IVの魅力と可能性

であれば閏年とする、など。

しかし、どのような方法で日付計算しても、これらの計算ロジックは完全に外部サブルーチン化してパラメーターを渡すような多機能なものではなく、プログラムごとに計算ロジックをコピーして流用し、Aのプログラムでは1週間後を算出、Bのプログラムでは1カ月後を算出、など、プログラムごとに機能が微妙に違ってきます。

結果としてバグが発生しやすく、とくに4年に1回の閏年には2月の計算処理が正しく行われるか、開発担当者の方々は当日を迎えるまで気が休まらないのではないのでしょうか。

## 日付フィールドの定義

RPG IVでは、これらの日付計算が演算命令や関数として用意され、データ構造もサブルーチンもいっさい必要ありません。演算命令の使用方法を理解しておけば、何

の迷いも悩みもなく日付計算が行われます。

日付計算を使用するには、計算対象のフィールドを、日付形式で定義しておく必要があります。データベースの場合、**図表2**のようにDDSのデータ・タイプに「L」を指定し、桁数、小数点以下の桁数は空白にしておきます。さらにDATFMTキーワードを使用し、パラメーターに日付形式を指定します。

また計算に使用できるのは、日付だけでなく、時刻も同様です。時刻形式のフィールドはデータ・タイプに「T」を指定します。時刻の形式にはTIMFMTキーワードを使用し、パラメーターに時刻形式を指定します。DATFMT、TIMFMTで使用できるパラメーターは**図表3**をご覧ください。

一方で、RPGプログラム内部で使用する可変フィールドでも、日付形式、時刻形式のフィールドを定義できます。

RPG IIIでの変数定義は演算仕様書で行いますが、変数に値を代入するロジックで、変数名、桁数、小数部の桁数を指定することによって定義します。小数部の桁数を指

図表2 RPG IVでの日付計算

FMT	PF	.....A.....	T.NAME+++++RLEN++TDPB.....	FUNCTIONS+++++
0012.00		A	TKGEND	9P 0 COLHDG (' 信用限度額 ')
0013.00		A	TKUZAN	9P 0 COLHDG (' 売掛金 ' ' 残高 ')
0014.00		A	TKNYUK	L DATFMT (*ISO)
0015.00		A		COLHDG (' 最終入金日 ')
0016.00		A	TKNYUT	T TIMFMT (*JIS)
0017.00		A		COLHDG (' 入金時間 ')

図表3 DATFMT、TIMFMTで指定するデータ・タイプ

日付のデータ・タイプ (DATFMT) 省略値は\*ISO

値	説明	形式	長さ	使用例
*MDY	月/日/年	MM/DD/YY	8	05/15/14
*DMY	日/月/年	DD/MM/YY	8	15/05/14
*YMD	年/月/日	YY/MM/DD	8	14/05/15
*JUL	通年	YY/DDD	6	14/135
*ISO	国際標準	YYYY-MM-DD	10	2014-05-15
*USA	USA標準	MM/DD/YYYY	10	05/15/2014
*EUR	ヨーロッパ標準	DD.MM.YYYY	10	15.05.2014
*JIS	日本工業規格	YYYY-MM-DD	10	2014-05-15

時刻のデータ・タイプ (TIMFMT) 省略値は\*ISO

値	説明	形式	長さ	使用例
*HMS	時.分.秒	HH:MM:SS	8	14:00:00
*ISO	国際標準	HH.MM.SS	8	14.00.00
*USA	USA標準	HH:MM AM	8	02:00 PM
*EUR	ヨーロッパ標準	HH.MM.SS	8	14.00.00
*JIS	日本工業規格	HH:MM:SS	8	14:00:00

定すれば数字フィールド、指定しなければ文字フィールドになりますので、変数とは数字か文字かのいずれかになってしまいます。

**図表4**をご覧ください。RPG IVからは定義仕様書（仕様書コード：D）が加わり、フィールドの定義のみならず、配列、データ構造もすべて定義仕様書で行えるように統一されました。定義仕様書ではデータ・タイプが指定できるので、日付形式のフィールドを定義する場合、データ・タイプに「D」を指定します（DDSでは「L」ですので、間違えないようにご注意ください）。時刻形式は、定義仕様書でも「T」を指定します。また、DATFMTキーワード、TIMFMTキーワードを指定できますが、使用方法はデータベースの場合と同じです。

定義仕様書では、INZキーワードを使用してフィールド

の初期値を指定できます。データベースや表示装置ファイルを作成する際にDDSでDFTキーワードが使用できますが、それと同じものと考えてよいでしょう。**図表4**にもあったように、RPG IVの定義仕様書では日付フィールドに対して、\*JOBを指定することにより、ジョブ日付を初期値として設定できます。また、\*SYSを指定するとシステム日付になります（時刻形式のフィールドで指定できるのは\*SYSのみです）。

## RPG IVの日付計算

以上の日付フィールドを使用して、日付、時刻の加算にはADDDUR命令、減算にはSUBDUR命令を使用しま

**図表4** RPGプログラムの可変フィールドでの日付形式・時刻形式の定義

```
FMT D  DNAME+++++++ETDSFROM+++TO/L+++IDC.KEYWORDS+++++++
0012.00 D @JOB                S                D  DATFMT(*JIS) INZ(*JOB)
0013.00 D @SYS                S                D  DATFMT(*JIS) INZ(*SYS)
0014.00 D @TIM                S                T  TIMFMT(*ISO) INZ(*SYS)
```

**図表5** 10日後の日付をNEXTDAYに入れる場合

```
@TODAY          ADDDUR      10:*D          @NEXTDAY
```

**図表6** 10カ月後の日付をNEXTDAYに入れる場合

```
@TODAY          ADDDUR      10:*M          @NEXTDAY
```

**図表7** 10年後の日付をNEXTDAYに入れる場合

```
@TODAY          ADDDUR      10:*Y          @NEXTDAY
```

**図表8** 時刻の計算

```
@NOW            SUBDUR      @STR_TIM      @HUN:*MN
```

## RPG IVの魅力と可能性

す。これらの命令が優れているところはパラメーターに\* Y、\* M、\* Dを指定することで、年、月、日、それぞれの単位で計算が行えます。

たとえば、フィールド TODAYには今日の日付が\* JIS形式で入っているとします。この日付から起算して、10日後の日付をNEXTDAYに入れたい場合は、**図表5**のように記述します。10カ月後の場合は**図表6**、10年後の場合は**図表7**のようになります。RPG IIIではこれらの計算ロジックに50~100ステップほど費やすことも多いかと思いますが、それがたった1ステップで可能になるのです。

時刻の場合も同様です。パラメーターに\* H、\* MN、\* Sを指定することで、時、分、秒の計算が行えます。こちらは減算と期間計算の例で紹介しましょう。

**図表8**をご覧ください。現在の時間@NOWから、処理開始時間@STR\_TIMを差し引き、その間が何分であったかが数値として@HUNに入ります。**図表9**は、期間コードの一覧です。

### 関数を使用した日付計算

ところで、前回ご説明したフリーフォーマットですが、ADDDUR、SUBDUR命令はフリーフォーマットでは使用できません。ADDDUR、SUBDUR命令だけでなく、フリーフォーマットでは使用できない演算命令がいくつかありますが、ほとんどの場合は関数で解決できます。関数を使うといっても、基本は+、-を使っただけの加算、減算です。

ポイントは、日数を関数に渡して使用することです。%YEARS関数に年を数値で渡すと、内部で日数変換され、期間計算に使用できます。月数は%MONTHS、日数は%DAYSを使用します(**図表10**)。**図表11**は今日の日付から20日後の日付を計算し、結果を@DATEに入れています。

### まとめ

このようにRPG IVでは日付形式、時間形式のフィールドを扱うことができ、期間計算は単純に加算、減算でしかありません。2020年2月28日の1週間後、などという場合も単純に7を足し算すればいいだけです。プログラムのステップ数を減らすことができ、プログラムの信頼度そのものも大幅に向上します。

どのシステム部門でも日付計算に関するロジックはサブ

ルーチン化されていたり、また、/COPYでステートメントに組み込んだり、ある種の伝統のようなものがあり、それがまた財産でもあると思います。それらを捨て去るのは非常に残念な気もしますが、この日付計算機能があるだけでも、RPG IIIからRPG IVに移行する価値があるというものです。

ここまで、フリーフォーマット、関数、日付計算と、RPG IIIにはないRPG IVの機能をご紹介しましたが、まだまだこんなものではありません。これからも機会があるごとに、RPG IVの魅力をお伝えしていきます。⑦

図表9 期間コード一覧

年	*YEARS または *Y
月	*MONTHS または *M
日	*DAYS または *D
時	*HOURS または *H
分	*MINUTES または *MN
秒	*SECONDS または *S
マイクロ秒	*MSECOND または *MS

図表10 数値の組み込み関数

%YEARS (数値)	年数を戻します。
%MONTHS (数値)	月数を戻します。
%DAYS (数値)	日数を戻します。
%HOURS (数値)	時間数を戻します。
%MINUTES (数値)	分数を戻します。
%SECONDS (数値)	秒数を戻します。
%MSECONDS (数値)	マイクロ秒数を戻します。

図表11 今日の日付から20日後の日付を計算し、結果を@DATEに入れる計算式

```
EVAL @DATE = @TODAY + %DAYS (20)
```