

## IEIEJ-G-0006:2006

# BACnet®システムインターオペラビリティガイドライン アデンダム a

## スケジュールオブジェクト, カレンダーオブジェクトの運用ガイド

BAS 標準インターフェース仕様推進拡張委員会

BAS 標準インターフェース仕様推進拡張委員会による決定:

規格 (IEIEJ - G - 0006 : 2006) の変更 (Change to Standard (IEIEJ - G - 0006 : 2006)):

**アデンダム a. スケジュールオブジェクト, カレンダーオブジェクトの運用ガイド**

(原規格条文に対して変更追加がわかる変更履歴付とする。取り消し線は 1 本線, 追加は斜体赤字で表現) 章, 節まとめて追加する場合は, 通常の手体で記述する。)

[4.2.5 スケジュールオブジェクト型(Schedule Object Type) 追加]

### 4.2.5 スケジュールオブジェクト型 (Schedule Object Type)

BACnet®-2004 では, スケジュールオブジェクトに関して, Present\_Value の演算方法が定義され, Schedule\_Default プロパティや Out\_Of\_Service プロパティが追加された。本項では, これらに対する解釈の相違により, 各デバイスの動作が異なるなど, インターオペラビリティを確保する上で問題となり得る部分について, ガイドラインとしての解釈を示すとともに, スケジュールオブジェクトの振る舞いを定義する。

#### 4.2.5.1 スケジュールオブジェクト関連の推奨値について

インターオペラビリティを確保するため, スケジュールオブジェクト関連のオブジェクト数, ARRAY 要素数などに関して, 推奨値を示す。

- 1) **スケジュールオブジェクトの数** スケジュールオブジェクトをサポートするコントローラが, コントローラあたりに持つべきスケジュールオブジェクトのインスタンス数を以下のように定義する。  
ひとつのコントローラにおいて,

スケジュールで操作可能なプロパティの数

≦ スケジュールオブジェクト数 \* 登録プロパティ数\*

という式が成り立つように, スケジュールオブジェクト数と各スケジュールに登録可能な登録プロパティ数を持つことが可能でなければならない。

※登録プロパティ数とは, ひとつのスケジュールオブジェクトあたり, オブジェクトプロパティの参照リスト(List\_Of\_Object\_Property\_Reference)プロパティに設定可能なリストの最大要素数のことを示す。

- 2) **Weekly\_Schedule プロパティと Exception\_Schedule プロパティのサポートについて** BACnet®-2004 では, 表 12-28. Schedule オブジェクト型のプロパティ群 や 12.24.7 週間スケジュール (Weekly\_Schedule) に, "Schedule オブジェクトの各々のインスタンスは Weekly\_Schedule か空でない Exception\_Schedule のいずれかをサポートするものとする"と示されており, Exception\_Schedule プロパティと Weekly\_Schedule プロパティのうち, どちらか1つを最低限サポートすることが要求されている。さらに, BACnet®-2004 の K.3.2 においては, "SCHED-B に適合する装置は少なくとも1つのカレンダーと1つのスケジュールオブジェクトを所有しなくてはならない"と示されており, Exception\_Schedule の ARRAY 要素の最低限ひ

とつをサポートすることが定められている。スケジュールオブジェクトが本来目的としている機能を実現すること、コントローラの自立性を確保しシステムの信頼性を向上させること、さらに、インターオペラビリティを確保することを目的として、当ガイドでは、スケジュールオブジェクトをサポートするコントローラは、`Weekly_Schedule` と `Exception_Schedule` の両方をサポートすることを推奨する。

- 3) **Exception\_Schedule の要素数について** システムにおけるスケジュールオブジェクトの運用において、`ARRAY` 型のプロパティである `Exception_Schedule` プロパティに関して、各コントローラでサポートする `ARRAY` の要素数が共通もしくは、ある値以上であることが望まれる。`Exception_Schedule` の `ARRAY` 要素数は、一般のビルの運用を考慮した上で、当ガイドでは 10 以上を推奨する。10 の根拠は、7 曜日+休日+2 特異日として使用することを想定した上で 10 としている。しかし、その実装においては、7 曜日+休日+2 特異日の使用に限定するものではない。また、`Exception_Schedule` の `ARRAY` 要素の個々の `Index` に対して、特定の意味づけを持たせるものではない。
- 4) **day-schedule, listOfTimeValues の要素数** `Weekly_Schedule` の `day-schedule` や `Exception_Schedule` の `listOfTimeValues` のリスト要素数に関して、BACnet®-2004 の **K.3.2** には、“スケジュールオブジェクトは 1 日に少なくとも 6 つの登録をサポートしなくてはならない”とある。当ガイドでは、規定のこの文章に従い、スケジュールオブジェクトをサポートするコントローラは、少なくとも 1 日あたり 6 つの要素数をサポートすることとする。BACnet®-2004 の規定では、`Weekly_Schedule` と、単一もしくは複数の `Exception_Schedule` の要素から、1 日のスケジュールを定義することが可能である。この BACnet®-2004 の **K.3.2** にある “6 つの登録”とは、ひとつずつの `Weekly_Schedule` の `day-schedule` の要素や `Exception_Schedule` の `listOfTimeValues` の要素数を示しているのではなく、1 日あたりの要素数を示していることに注意しなければならない。

#### 4.2.5.2 Present\_Valueプロパティの演算とコマンド出力について

スケジュールオブジェクトの `Present_Value` の演算については、BACnet®-2004 の **12.24.4** に示されており、詳細はここには定義しない。しかし、コマンド出力のタイミングに関しては、明確な記述が存在しない。当ガイドでは、解釈の統一を図るため、コマンド出力のタイミングを以下のように定める。

- (a) `BACnetTimeValue` に指定されている時刻にコマンド出力をする。スケジュールオブジェクトの `Present_Value` と `BACnetTimeValue` の `Value` に変化がなかった場合でも、`BACnetTimeValue` に設定された時刻には、コマンドを出力する。但し、`BACnetTimeValue` の `time` が 00:00 の場合は、スケジュールオブジェクトの `Present_Value` と `BACnetTimeValue` の `value` が同じ時はコマンドは出力しない。
- (b) `Present_Value` の演算の結果、`Schedule_Default` を出力する場合は、`Present_Value` の値が変化したときのみコマンド出力をする。

BACnet®-2004 では、`BACnetTimeValue` に含まれるコマンドの値として、`NULL` を定義している。この `NULL` のコマンドと `Schedule_Default` プロパティに関して、当ガイドでの解釈を以下に示す。

- 当ガイドでは、システムの制御を適切に設計できることを目的として、`Schedule_Default` に `NULL` を設定してはならないものとする。その理由を以下に示す。

`Exception_Schedule` の `BACnetTimeValue` に時刻とともに `NULL` が指定されていた場合、その指定された時刻に、該当の `ARRAY` 要素は、出力コマンドの候補から外れることを意味する。`NULL` とともに、指定された時刻の出力コマンドの演算には、次に優先される `ARRAY` 要素、`NULL` だった場合は `Weekly_Schedule` の設定、これも `NULL` なら最終的には、`Schedule_Default` の設定が出力される。`Schedule_Default` が `NULL` であった場合、スケジュールオブジェクトからのコマンド出力が `NULL` となる。スケジュールオブジェクトからの出力が `NULL` となった場合、他のアプリケーションからの出力が有効になるか、もしくは、`Relinquish_Default` が有効になり、設備に対して、その値がコマンド出力される。システムのコマンドプライオリティの設計によっては、このような動作は、予期せぬ動作を引き起こすこととなり、制御機能の設計が管理しきれないものとなる。以上の理由から、`Schedule_Default` に `NULL` を設定する運用を禁止する。

- Weekly\_Schedule と Exception\_Schedule との関係において、例えば Weekly\_Schedule で 8:30 ON, 17:00 OFF となっているときに Exception\_Schedule で 10:00 ON 18:00 OFF としたいケースを考えると、このケースを実現するには Exception\_Schedule で 8:30 以前に OFF という設定をしておく必要がある。この 8:30 以前の OFF は 8:30 以前に Exception\_Schedule を有効にするためのもので、通常はその日は Exception\_Schedule で運用するというので、Exception\_Schedule で 00:00 OFF とする運用が考えられる。(a) によりこのような運用を可能とする。
- Scheule\_Default により、Present\_Value が決まるケースでは、Present\_Value の値が変化した場合のみ、コマンド出力を行うこととする。これによって特に 00:00 の不要なコマンドの出力、それに伴う Notification 等の不要なメッセージを抑制する。
- 00:00 をまたいで継続し、かつ 00:00 に出力されているべき値が Schedule\_Default の値と異なる値のスケジュールを組みたい場合、規定上、00:00 の TimeValue が必ず存在しなければならない。しかし、そうすると(b)に記述した通り、00:00 に必ず出力が発生することになり、本来、実施したい動作とはことなる。また、00:00 に予期せぬコマンドが出力される可能性があり、これは、ビルの運用上、問題となる可能性がある。日替わりをまたいだスケジュールを実現するためには、コントローラ側で下記の Schedule\_Default の振る舞いを可能とすることを推奨する。  
Schedule\_Default は R(Required)なので、必ずしも Writable である必要はない。Read Only とし、コントローラ内で、Schedule オブジェクトの PV が変化する都度 Schedule\_Default を、その PV の値にすれば、Schedule\_Default の目的は達成でき、00:00 に予期せぬコマンドが発生することも回避できる。

#### 4.2.5.3 Exception\_ScheduleとWeeklyScheduleの書込みについて

Weekly\_Schedule プロパティ、Exception\_Schedule プロパティへの書込みは、プロパティをすべて書込みするか、Index を指定して書込みをするか、どちらでも問題はないが、セグメンテーションの可能性を考慮すると、Index を指定して書込みをするのが望ましい。当ガイドでは、Index を指定して書込みを行うことを推奨する。

#### 4.2.5.4 Exception\_Scheduleの Periodについて

Exception\_Schedule の Period プロパティは、該当する日付、日付の範囲、曜日、もしくは、カレンダーオブジェクトの参照を定義可能である。日付や日付の範囲を指定していた場合、該当する日付はすでに過去のものとなっており、システムの日付が変更されない限り、その Exception\_Schedule の要素が無意味となることが考えられる。無意味となった Exception\_Schedule の要素の削除は、B-OWS が行うこととする。日付の変更の可能性などにより、対象の Exception\_Schedule が意味のあるものとなる可能性もあるため、スケジュールオブジェクトをサポートするコントローラでは、削除を行わない。日付や時刻を管理する B-OWS がそのような Exception\_Schedule の要素の削除を行う。

#### 4.2.5.5 List\_Of\_Object\_Property\_Referenceプロパティについて

BACnet@-2004 では規格上は全てのオブジェクトの全ての書込み可能なプロパティは List\_Of\_Object\_Property\_Reference の対象にできる。また、BTL ガイドラインには“10.3 Schedule objects should not be used to schedule complex or proprietary data types.”という項目があり、少なくとも基本データ型のプロパティは対象にできる。しかし、これはすべての基本データ型プロパティのサポートを必須とするという意味ではなく、どれをサポートするかはベンダーマターである。ビルの運用上、最低限必要と考えられる BO, BV, MO, MV の Present\_Value を対象とすることを推奨する。

また、単一のプロパティを複数のスケジュールオブジェクトの List\_Of\_Property\_Reference に登録することは、B-OWS でのアプリケーションを考慮したとき、各設備のスケジュール設定を表現する上で、処理が煩雑となることから、避けることが望ましい。

#### 4.2.5.6 Out\_Of\_Serviceプロパティについて

スケジュールオブジェクトの設定情報を保持したまま、一時的にスケジュール動作からの除外/再登録について、この Out\_Of\_Service プロパティを使用する。規定上 Writable である必要はないが、一時的にスケジュー

ールを動作させないようにする機能はビルの運用上、または調整上、必要となることが多い。Out\_Of\_Service は Writable とすることを推奨する。また、Out\_Of\_Service が TRUE であるときの動作は、BACnet@-2004 の 12.24.14 に示されるとおりとする。以下にその内容を抜粋する。

”論理値型の Out\_Of\_Service プロパティは、スケジュールオブジェクトの内部演算が、Present\_Value プロパティの値を決定する為に使用される(TRUE/FALSE)か、されない(FALSE/TRUE)かを示す。これが意味するところは、Out\_Of\_Service の値が TRUE である場合、Present\_Value プロパティは内部演算から切り離され、Present\_Value プロパティが他のプロパティの変化に追従しないということである。Out\_Of\_Service が TRUE のとき、List\_Of\_Object\_Property\_Reference のメンバへ値を書込むなど Present\_Value に依存する他の機能は、まるで Present\_Value の変化が内部演算により発生したかのように、その変化に応じて反応する。”

#### 4.2.5.7 Priority\_For\_Writingプロパティについて

スケジュールオブジェクトのコマンド出力時のコマンド優先順位だけではなく、ひとつのシステムにおいて、コマンド優先順位は統一されていなければ、制御機能の設計は不可能である。BACnet@-2004 の Table 19-1 に、標準のコマンド優先順位が定義されているが、ここには、スケジュールオブジェクトのコマンド出力時の優先順位が定義されていない。当ガイドラインでは、インターオペラビリティを確保する目的で、スケジュールオブジェクトの Priority\_For\_Writing の値を 8 とする。BACnet@-2004 では、Manual Operator のコマンド優先順位を 8 としているが、これと同じ値とする。オペレータのマニュアル操作の後、スケジュールからの出力により、機器が動作する必要があること、スケジュールからの出力の後、オペレータのマニュアル操作により、機器が動作する必要があることから、Manual Operator のコマンド優先順位と同じ値を推奨する。

#### 4.2.5.8 延長運転課金機能とスケジュールオブジェクトについて

延長運転課金機能とスケジュールオブジェクトの運用について、記述する。延長運転課金機能を実現する B-OWS ベンダの、実現方法、考え方に依存するので、一意の実現方法をガイドラインで定めることには、無理があると判断されるため、ここでは、運用例を示すにとどめる。

これまで伝統的にコアタイムスケジュールとそれに対する追加運転という概念のベンダ以外に、コアタイムスケジュールが変更となるケースを休日スケジュールと休日カレンダー、特異日 1 スケジュールと特異日 1 カレンダー、特異日 2 スケジュールと特異日 2 カレンダーという考え方で実現してきたベンダも存在する（特異日をいくつ持つかはベンダによって異なる）。さらに、カレンダーはシステムで一括管理する必要があるので、マスタは B-OWS が管理し、必要としているコントローラに展開する必要がある。

以上の状況を踏まえて、以下は規定上運用の範囲とみなされるので、運用例として記述する。

##### 運用例 1

Weekly\_Schedule をコアタイムスケジュールとして、この Device にとっての休日カレンダー NO. 特異日 1 カレンダー NO., 特異日 2 カレンダー NO. を予め定めて、Exception\_Schedule で CalendarReference で指定した NO.で、休日、特異日 1、特異日 2 として扱う。

##### 運用例 2

残業延長機能は全て OWS で行う。特異日という概念は B-OWS で管理する。

## [4.2.6 カレンダーオブジェクト型(Calendar Object Type) 追加]

## 4.2.6 カレンダーオブジェクト型

本項では、4.2.5. に示すスケジュールオブジェクトについての解釈に従い、インターオペラビリティを確保する上で、必要となると考えられるカレンダーオブジェクトの制限事項について、記述する。

## 4.2.6.1 カレンダーオブジェクト関連の推奨値について

- 1) コントローラあたりのカレンダーオブジェクト数について B-BC 毎に、サポートすべきカレンダーオブジェクト数は、

スケジュールで操作可能なプロパティ数

/ 1 スケジュールあたりに登録可能なプロパティ数

\* 特異日の種類(休日を含む)

で計算される。例えば、スケジュールで操作可能なプロパティの数が 1,000 であるコントローラにおいて、ひとつのスケジュールあたりに登録可能なプロパティ数が 30 オブジェクト、特異日の種類(休日を含む)が 3 種類であった場合、サポートすべきカレンダーオブジェクトの数は、100 となる。

- 2) カレンダーオブジェクトあたりの Date\_List プロパティの要素数について 実際のシステムにおいては、セグメンテーションをサポートしているデバイス、サポートしていないデバイスが混在する可能性が考えられ、また、セグメンテーションのサポートは、必須でないことを考慮し、当ガイドラインでは、Date\_List の要素数は、ReadPropertyMultiple, AddListElement, RemoveListElement などの通信メッセージにおいて、Max\_APDU\_Length を超えない範囲とする。ネットワーク層のヘッダの長さにも依存するが、Date\_List の要素のデータ型が、Date であった場合、1 日あたり 5 バイトの長さで定義可能であり、約 200 要素(200 日)を設定可能であり、先 1 年間のカレンダー設定を行うことを想定すると、十分な日数であると判断できる。200 日以上を設定したいケースがないとは言いきれないが、このようなケースが発生した場合、Exception\_Schedule の要素を複数使用して、複数のカレンダーオブジェクトにより、機能を実現可能である。

## 4.2.6.2 カレンダーのマスタについて

BACnet@-2004 規定においては、カレンダーオブジェクトは、スケジュールオブジェクトの Exception\_Schedule の Period で参照されることを想定して定義されており、カレンダーオブジェクトの各プロパティの設定は、コントローラ毎、スケジュールオブジェクト毎となっているが、実際のビルの運用においては、カレンダーの設定は、複数のコントローラをまたいでビル全体で共通であるケースが多いと考えられる。当ガイドラインでは、基本的に、システムとして、一括でカレンダーを管理するデバイスが存在するものとする。また、機能分散の観点から、カレンダーを一括で管理するデバイスは、各コントローラへカレンダーオブジェクトの設定を展開する機能を持つ必要がある。

## 4.2.6.3 カレンダーオブジェクトのDate\_List要素のうち、経過した日付の要素の扱いについて

カレンダーオブジェクトの Date\_List の要素は、該当する日付がすでに過去のものとなっており、システムの日付が変更されない限り、その日付の要素が無意味となっているものが考えられる。無意味となった Date\_List の要素の削除は、B-OWS が行うこととする。日付の変更の可能性などにより、対象の Date\_List が意味のあるものとなる可能性もあるため、カレンダーオブジェクトをサポートするコントローラでは、削除を行わない。日付や時刻を管理する B-OWS がそのような Date\_List の要素の削除を行う。