

Windows 対応 BACnet ドライバ DLL

ユーザーズマニュアル (Version 2.2.1.0 以上)



文書番号	HM000054-D
初版発行日	2021 年 4 月 8 日
改定日	2023 年 12 月 28 日

<https://www.uni-tec.co.jp/>

はじめに

この度は、株式会社ユニテック製 Windows 対応版 BACnet 通信ドライバ DLL をお買い上げいただきまことにありがとうございました。

ご使用前に BACnet の仕様・機能・性能を十分ご理解のうえ、本書をよくお読みいただき、正しくご使用いただけますようお願いいたします。

また、BACnet 上に接続する弊社製品以外への接続性、損傷およびその他の業務に対する保証について、株式会社ユニテックは一切の責任を負いません。現場での正式使用の前には、お客様の責任として十分な接続試験を行って検証していただけますようお願いいたします。

なお、本製品によって生じた機会損失、逸失利益、当社の予見の有無を問わず特別の事情から生じた損害、二次災害、事故補償およびその他の業務に対する保証について、株式会社ユニテックは一切の責任を負いません。

おことわり

- (1) 本書の内容の一部または全部を無断で複写、複製、転載することを禁じます。
- (2) 本書の内容に関しては改善のため、予告なしに仕様などを変更することがありますので、予めご了承ください。
- (3) 本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがありましたら、お手数ですが巻末記載の弊社までご連絡ください。
- (4) 本書によって、産業財産権やその他の権利の実施に対する保証、または実施権を許諾するものではありません。また、本書の内容により起因する産業財産権上の諸問題について、株式会社ユニテックは一切の責任を負いません。

商標について

- BACnet は American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) の商標です。
- MELSEC、MELSEC iQ-R は、三菱電機株式の登録商標です。
- Windows、Windows 11、Visual Studio、Visual C++、Visual Basic は、米国およびその他の国々での Microsoft 社の登録商標です。
- その他、各社名・各サービス名・システム名・製品名は、各社・各組織の商標または登録商標です。

改定履歴

改定日付	改定内容	副番
2021 年 4 月 8 日	初版発行	
2021 年 4 月 28 日	「1.8 ご提供の方法とサポート」を追記 「3.7.105BD_ReleaseHandle」の参考に注意事項を追記	A
2022 年 5 月 30 日	下記対応規格の追加に伴い、全体を追記・修正 <ul style="list-style-type: none"> ● ANSI/ASHRAE Standard 135-2012 ● IEC 60335-1:2017 下記通信プロトコルの追加に伴い、全体を追記・修正 <ul style="list-style-type: none"> ● BACnet/IP v6 対応規格の追加に伴い、構成ファイルに下記のパラメータを追加 <ul style="list-style-type: none"> ● AnnexUEnable ● SystemStatusReadInterval ● CheckAliveInterval BACnet/IP v6 に伴い、下記の関数に AnnexUEnable 引数を追加 <ul style="list-style-type: none"> ● 「3.7.3BD_OpenDirect」 ● 「3.7.4BD_OpenDirectB」 「2.4 サポート BIBB」を BACnet2012 規格基準に変更	B
2022 年 7 月 5 日	あいまいな仕様の明確化 「5.2 保証について」の一部文言を修正 「商標について」において、BACnet は商標であることを記載	C
2023 年 12 月 28 日	ExceptionSchedule 7 日分を 1 パケットで読み書きできる以下の関数を追加 「3.7.69 BD_WriteExceptionScheduleForSevenDays」 「3.7.70 BD_ReadExceptionScheduleForSevenDays」 「3.6 関数一覧」に上記関数を追記 「3.7.2 BD_Open」の「3.7.2(2) デバイス管理対象リストファイル」の CSV ファイルの仕様を変更 表紙の変更 コピーライトを 2023 に更新 「5.2 保証について」の内容を修正 「5.3 サポートについて」の内容を修正	D

© 2023 UNITEC CORPORATION

目次

第1章 概要.....	9
1.1. システム構成.....	10
1.2. 特長.....	11
1.3. ご用意いただく物.....	12
1.4. 使用環境.....	12
1.5. ご利用上の注意.....	13
1.6. 関連製品のご案内.....	14
1.7. BACnet について	15
1.8. ご提供の方法とサポート	19
第2章 仕様.....	20
2.1. BACnet 規格	20
2.2. BACnet 通信	20
2.3. サポートオブジェクト	21
2.4. サポート BIBB	22
2.5. オブジェクトプロパティ	24
2.5.1. Analog Input Object Type (AI)	25
2.5.2. Analog Output Object Type (AO)	26
2.5.3. Analog Value Object Type (AV)	27
2.5.4. Binary Input Object Type (BI)	28
2.5.5. Binary Output Object Type (BO)	29
2.5.6. Binary Value Object Type (BV)	30
2.5.7. Multi-state Input Object Type (MI)	31
2.5.8. Multi-state Output Object Type (MO)	32
2.5.9. Multi-state Value Object Type (MV)	33
2.5.10. Accumulator Object Type (AC)	34
2.5.11. Calendar Object Type (CA)	35
2.5.12. Group Object Type (GR)	35
2.5.13. Notification Class Object Type (NC)	35
2.5.14. Schedule Object Type (SC)	36
2.5.15. Trend Log Object Type (TL)	37
2.5.16. 計量オブジェクト	38
2.5.17. 電力デマンド監視オブジェクト	39
2.5.18. 電力デマンド制御オブジェクト	41
2.5.19. 発電機負荷制御オブジェクト	42
2.5.20. Device Object Type (DV)	43
2.6. オプションプロパティ	45
2.7. SystemStatus プロパティ.....	46
第3章 機能.....	47
3.1. オブジェクト作成機能	47
3.2. プロパティアクセス機能	47
3.3. 定期収集機能.....	47
3.4. BACnet デバイス管理機能.....	47
3.5. コールバック関数機能	48
3.6. 関数一覧.....	49

3.7. 関数仕様.....	52
3.7.1. 関数共通の仕様.....	52
3.7.2. BD_Open.....	55
3.7.3. BD_OpenDirect.....	59
3.7.4. BD_OpenDirectB.....	60
3.7.5. BD_Close.....	60
3.7.6. BD_AddMonitorDevice.....	61
3.7.7. BD_AddRestartNotificationDevice.....	61
3.7.8. BD_AddRestartNotificationAddress.....	61
3.7.9. BD_AddScan.....	62
3.7.10. BD_AddScanGroup.....	62
3.7.11. BD_AddRemoteDevice.....	63
3.7.12. BD_SetDeviceStatusChangeCallback.....	63
3.7.13. BD_SetAccessCallback.....	64
3.7.14. BD_SetScanCallback.....	64
3.7.15. BD_SetScanPropertyCallback.....	65
3.7.16. BD_SetCOVCallback.....	65
3.7.17. BD_SetEventCallback.....	66
3.7.18. BD_PropertyChangeCallback.....	66
3.7.19. BD_GetDeviceStatus.....	67
3.7.20. BD_GetDeviceAddressBinding.....	67
3.7.21. BD_SendWhois.....	68
3.7.22. BD_SendWhoisWithRange.....	68
3.7.23. BD_SendIam.....	68
3.7.24. BD_GetMyDeviceInstanceNo.....	69
3.7.25. BD_StartStopScan.....	69
3.7.26. BD_ReadScanData.....	70
3.7.27. BD_ReadScanMultipleData.....	70
3.7.28. BD_SetSystemStatus.....	71
3.7.29. BD_SetMaxOutStandingPDUCount.....	71
3.7.30. BD_SetWhoIsRangeSendInterval.....	72
3.7.31. BD_SendTimeSync.....	72
3.7.32. BD_SendTimeSyncUnicast.....	72
3.7.33. BD_SendUTCTimeSync.....	73
3.7.34. BD_SetSubscribeCOV.....	73
3.7.35. BD_WriteProperty.....	74
3.7.36. BD_WritePropertyUnsigned.....	75
3.7.37. BD_WritePropertyEnumerated.....	76
3.7.38. BD_WritePropertyReal.....	77
3.7.39. BD_WritePropertyBool.....	78
3.7.40. BD_WritePropertyCharacterString.....	79
3.7.41. BD_WritePropertyEventTransitionBits.....	80
3.7.42. BD_WritePropertyLimitEnable.....	81
3.7.43. BD_WritePropertyMultiple.....	81
3.7.44. BD_WritePropertyMultipleA.....	82
3.7.45. BD_ReadProperty.....	83
3.7.46. BD_ReadPropertyUnsigned.....	83
3.7.47. BD_ReadPropertyEnumerated.....	84

3.7.48.	BD_ReadPropertyReal	84
3.7.49.	BD_ReadPropertyUnsignedWithStatus.....	85
3.7.50.	BD_ReadPropertyEnumeratedWithStatus.....	86
3.7.51.	BD_ReadPropertyRealWithStatus	87
3.7.52.	BD_ReadPropertyBool	87
3.7.53.	BD_ReadPropertyCharacterString	88
3.7.54.	BD_ReadPropertyEventTransitionBits.....	89
3.7.55.	BD_ReadPropertyLimitEnable	89
3.7.56.	BD_ReadPropertyStatusFlags	90
3.7.57.	BD_ReadPropertyMultiple	91
3.7.58.	BD_ReadPropertyMultipleA	91
3.7.59.	BD_AddListElement	92
3.7.60.	BD_RemoveListElement	92
3.7.61.	BD_WriteScheduleObjectPropertyList.....	93
3.7.62.	BD_ReadScheduleObjectPropertyList.....	93
3.7.63.	BD_AddScheduleObjectPropertyList.....	94
3.7.64.	BD_RemoveScheduleObjectPropertyList.....	94
3.7.65.	BD_WriteWeeklySchedule	95
3.7.66.	BD_ReadWeeklySchedule	95
3.7.67.	BD_WriteExceptionSchedule	96
3.7.68.	BD_ReadExceptionSchedule	97
3.7.69.	BD_WriteExceptionScheduleForSevenDays.....	98
3.7.70.	BD_ReadExceptionScheduleForSevenDays.....	99
3.7.71.	BD_WriteScheduleEffectivePeriod.....	100
3.7.72.	BD_ReadScheduleEffectivePeriod	100
3.7.73.	BD_WriteDateList	101
3.7.74.	BD_ReadDateList	101
3.7.75.	BD_AddDateList	102
3.7.76.	BD_RemoveDateList	102
3.7.77.	BD_ReadRangeTrendLogByPosition	103
3.7.78.	BD_ReadRangeTrendLogBySequenceNo.....	104
3.7.79.	BD_ReadRangeTrendLogByTime	105
3.7.80.	BD_AddObjectID	106
3.7.81.	BD_SetProperty	106
3.7.82.	BD_GetProperty	106
3.7.83.	BD_SetPropertyUnUsed	107
3.7.84.	BD_SetPVAndReliabilityUnsigned	107
3.7.85.	BD_SetPVAndReliabilityBinary	108
3.7.86.	BD_SetPVAndReliabilityReal	108
3.7.87.	BD_SetFeedBackValueAndReliabilityUnsigned.....	109
3.7.88.	BD_SetFeedBackValueAndReliabilityBinary.....	109
3.7.89.	BD_SetReliability	110
3.7.90.	BD_AddPulse	110
3.7.91.	BD_SetPropertyCharacterString	110
3.7.92.	BD_GetPropertyCharacterString	111
3.7.93.	BD_SetPropertyAlarmValues	111
3.7.94.	BD_GetPropertyAlarmValues	112
3.7.95.	BD_SetPropertyFaultValues	112

3.7.96.	BD_GetPropertyFaultValues	113
3.7.97.	BD_SetPropertyRecipientList	113
3.7.98.	BD_SetPropertyLogDeviceObjectPropertyReference	114
3.7.99.	BD_SaveBACnetObject	114
3.7.100.	BD_SaveBACnetObjectAll	114
3.7.101.	BD_SaveObjectCSVFile	115
3.7.102.	BD_LoadBACnetObject	115
3.7.103.	BD_ReinitializeDevice	115
3.7.104.	BD_DeviceCommunicationControl	116
3.7.105.	BD_ReleaseHandle	116
3.8.	データ構造	117
3.8.1.	tReturnCode	117
3.8.2.	tSimpleBitString	117
3.8.3.	tSimpleCharacterString	117
3.8.4.	tSimpleValue	118
3.8.5.	tLimitEnable	118
3.8.6.	tEventTransitionBits	118
3.8.7.	tStatusFlags	118
3.8.8.	tPropertyValue	119
3.8.9.	tWriteAccessSpec	119
3.8.10.	tObjectProperty	119
3.8.11.	tTime	119
3.8.12.	tTimeValue	119
3.8.13.	tDate	120
3.8.14.	tDateTime	120
3.8.15.	tDateRange	120
3.8.16.	tWeekNDay	120
3.8.17.	tPeriod	121
3.8.18.	tCalendarEntry	121
3.8.19.	tCOVData	121
3.8.20.	tTimeStamp	121
3.8.21.	tNotificationChangeOfState	122
3.8.22.	tNotificationChangeOfValue	122
3.8.23.	tNotificationCommandFailure	122
3.8.24.	tNotificationFloatingLimit	122
3.8.25.	tNotificationOutOfRange	123
3.8.26.	tNotificationBufferReady	123
3.8.27.	tNotificationUnsignedRange	123
3.8.28.	tEventData	124
3.8.29.	tLogRecord	124
3.8.30.	tAddressBinding	124
3.8.31.	tAlarmSummary	125
3.8.32.	tEnrollmentSummary	125
3.8.33.	tWHDData	125
3.8.34.	tScale	125
3.8.35.	tPrescale	125
3.8.36.	tRecipient	126
3.8.37.	tDayOfWeek	126

3.8.38. tRecipientList	126
3.8.39. tLogDeviceObjectProperty	126
3.8.40. tNotificationChangeOfReliability.....	127
第4章 プログラミング.....	128
4.1. 関数の使用例.....	128
4.1.1. 参入離脱の例.....	128
4.2. 他のデバイスのカレンダーオブジェクトの DateList の読み込み.....	129
4.3. 自身のオブジェクト作成.....	131
第5章 トラブルシューティングと保証.....	132
5.1. トラブル状況と対策.....	132
5.2. 保証について.....	133
5.3. サポートについて.....	135

第1章 概要

本マニュアルは、Windows 対応版 BACnet 通信ドライバ DLL（以下「本製品」または「本ドライバ」と略す）の仕様および取扱い方法などについて説明するものです。

本製品は、電気設備学会 IEIEJ-G-0006:2006 の B-BC ランクの機能を有しており、BACnet へ接続できるオペレータワークステーション（以下「OWS」と略す）やビルディングコントローラ（BC）を開発する際に使用できる Windows 用のソフトウェアです。

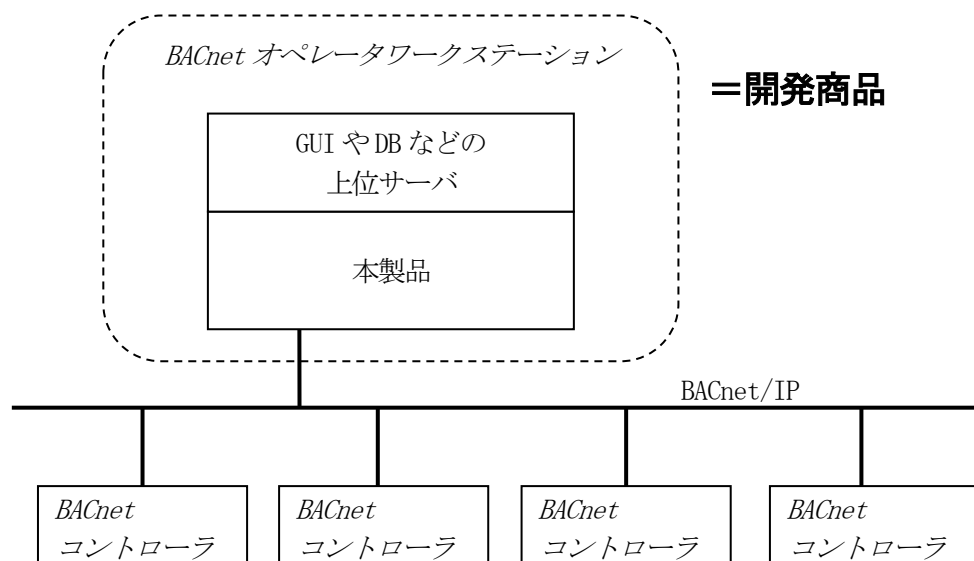
これにより、BACnet オペレータワークステーションや BACnet コントローラをお客様にて容易に開発することができます。

（以下、このお客様にて開発する部分を含め「開発商品」と略す）

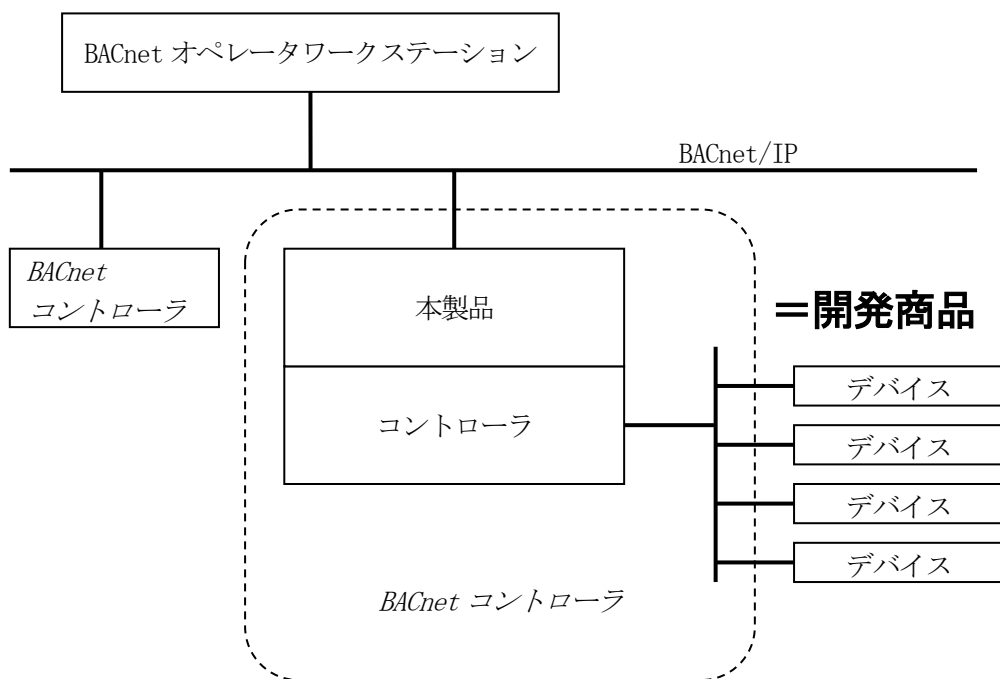
なお、お客様によっては、契約やカスタマイズ対応により、本書に記載した内容のすべてを満たしていない場合や、機能拡張をされている場合があります。契約やカスタマイズ対応時の資料、仕様を必ず保管し、その内容を合わせてご確認ください。

1. 1. システム構成

(1) BACnet オペレータワークステーションの開発



(2) BACnet コントローラの開発



1.2. 特長

(1) Visual C++でBACnet オペレータワークステーション開発

本製品は、DLL (DynamicLinkLibrary) として提供されます。これにより難しい BACnet の仕様をすべて理解することなく、BACnet 通信に対応した商品を Visual C++で開発することができます。

(2) 様々な BACnet 規格を選択

各種 BACnet 規格に準拠しており、自由に開発することができます。対応できる BACnet 仕様は次の通りです。

- ANSI/ASHRAE Standard 135-2004
- ANSI/ASHRAE Standard 135-2010
- ANSI/ASHRAE Standard 135-2012
- IEIEJ-P-0003:2000
- IEIEJ-P-0003:2000 アデンダム a
- IEIEJ-G-0006:2006 アデンダム a
- IEIEJ-G-0006:2017

(3) 通信プロトコルを選択

通信プロトコルを選択することができます。

- BACnet/IP (IPv4/IPv6 アドレス)
- BACnet/IPv6

(4) BACnet コントローラやBACnet オペレータワークステーションの構築を容易にする高機能

本製品は、電気設備学会 IEIEJ-G-0006:2006 の BACnet 仕様に準拠した B-BC 相当の仕様となっており、容易に BACnet システムを構築するために次の優れた機能を持っています。

- プロパティアクセス機能
- デバイス管理機能
- データ定期収集機能

(5) 高機能 BACnet オペレータワークステーションの開発を実現

Schedule オブジェクトや Calendar オブジェクト、TrendLog オブジェクトに対応しており、高度な BACnet 対応商品の開発を容易にします。

また、定期収集の機能を備えており、BACnet オペレータワークステーションの開発コストを抑えることができます。

1.3. ご用意いただく物

本製品のご利用にあたっては、別途、次の物を必要に応じて事前にご用意ください。

品名	数量	用途/詳細
開発商品パソコン	1 式	本製品をインストールし動作させるもの。
開発用環境	1 式	本製品を使用し製品を開発する環境。 <ul style="list-style-type: none">● 上位サーバなどのシステム● Visual C++などの開発ツール● 開発者用の開発商品パソコン
LAN ケーブル	1 本	BACnet システムと接続するために使用。

ポイント BACnet 接続の確認

BACnet によるネットワークの構築は、さまざまな点で障害が起こります。

株式会社ユニテック製の開発支援ツールを合わせてご購入いただくことをお勧めいたします。

1.4. 使用環境

本製品のご利用にあたっては、次の開発商品パソコンや開発用環境をご用意ください。

項目	動作環境
OS	Windows 8.1 / 10 / 11 Windows Server 2012 / 2012 R2 / 2016 / 2019 / 2022
開発環境	Visual Studio 2008、2012、2015、2019
開発言語	C++

注意：DLL での提供ですが、Visual Basic での動作に問題があるため、Visual Basic では使用することができません。

1.5. ご利用上の注意

(1) 利用するパソコンについて

- 本製品は、**Windows** で動作する **DLL** であり、パソコンおよびケーブル類に関しては、お客様にてご用意していただく必要があります。
- 本製品を利用するパソコンの推奨性能に関しては、「1.3. ご用意いただく物」を参照ください。
- 本製品は、パソコン上に不要なソフトウェアがインストールされていないことを前提としています。パソコン上にインストールされているソフトウェアやドライバによっては、本製品または他のソフトウェアが正常に動作しないこと、性能上に問題が生じることなどがあります。なお、当社以外のソフトウェアとの組み合わせに関する動作保証、動作確認、動作調査に関して、当社は一切の対応を行いません。

(2) 性能について

- 本製品は、**Windows** パソコン上で動作しますので、パソコンの性能や**OS**のバージョンによって性能差が発生します。
- オブジェクト点数や通信状況によっては、十分な性能を実現できないことがあります。

(3) 著作権や商標

- 本製品の著作権、および産業財産権はすべて株式会社ユニテックのものであり、ソースの公開などはい行いません。
- 開発商品において、株式会社ユニテックの社名、および本製品のロゴ・製品名・商標の利用は原則として行いません。使用を希望する場合は事前の書面による同意を得ていただけますようお願いいたします。

※本書末の「5.2. 保証について」「5.3. サポートについて」もあわせて参照ください。

1.6. 関連製品のご案内

本製品のご利用もしくはご検討の際には、次の当社製品についてもご検討ください。

(1) BACnet オペレータワークステーションを開発される場合

製品名	概要
BACnet Testing Tool - Controller	BACnet オペレータワークステーション側の試験を行うために、BACnet コントローラ側の機能を提供する Windows アプリケーションソフトです。 開発製品の試験を行う際には威力を発揮します。
MELSEC-Q シリーズ対応 BACnet インタフェースユニット 形名：BAQ08V	三菱電機株式会社製の MELSEC-Q シリーズに対応した BACnet 通信ユニットです。 開発製品と組み合わせて BACnet オペレータワークステーションを容易に開発することができます。
MELSEC iQ-R シリーズ対応 BACnet インタフェースユニット 形名：BAR16V	三菱電機株式会社製の MELSEC iQ-R シリーズを BACnet コントローラとして活用できる制御ユニットです。 シーケンサに直接装着できるため、非常に高速かつ安定したデータ更新が可能です。各種 BACnet 規格に準拠しており、ビル設備の監視制御実現を強力にサポートします。 開発製品と組み合わせて BACnet オペレータワークステーションを容易に開発することができます。

(2) BACnet コントローラを開発される場合

製品名	概要
BACnet Testing Tool - Monitor	BACnet コントローラ側の試験を行うために、BACnet オペレータワークステーション側の機能を提供する Windows アプリケーションソフトです。 開発製品の試験を行う際には威力を発揮します。

1.7. BACnet について

本書では紙面の関係上、BACnet および BACnet プロパティの詳細な説明は記載しておりません。BACnet および BACnet プロパティの詳細は、BACnet 仕様書にてご確認ください。

本製品をご使用する際には、BACnet に関する十分な知識を得ている必要はありませんが、本書は十分理解した上でご使用いただきますよう、お願いいたします。

本書では、BACnet システムの下位側を「BACnet コントローラ」、上位側を「BACnet オペレータワークステーション」と記述します。

BACnet に関する詳細をお知りになりたい方は、BACnet 仕様書を参照してください。

【BACnet 仕様書】

発行：電気設備学会 『BACnet ビルディングオートメーション用データ通信プロトコル』
電気設備学会 『BACnet システムインターオペラビリティガイドライン』

(1) BACnet の仕組み

BACnet は通常、照明・空調・防犯・防災・駆動設備などが直接または間接的に接続されている下位側と、これらを管理・監視・制御する上位側のシステムに分けることができます。

下位側を「B-BC」と称することが多く、「BACnet デバイス」または「ICONT」とも称します。上位側を「B-OWS」と称し、「中央監視」「HIM」などとも称します。

BACnet コントローラは、いくつかの BACnet オブジェクトを持っており、オブジェクトタイプによっては、アナログ値やデジタル値の入出力として照明・空調・防犯・防災・駆動設備などと直接または間接的に接続されます。そして、それぞれの BACnet オブジェクトは、使用目的や存在理由などから、さまざまな情報、つまり属性を保持しており、この情報のことをプロパティと称します。

BACnet オペレータワークステーションと BACnet コントローラ間、または BACnet コントローラ双方間では、この BACnet オブジェクトのプロパティなどを相互でリード/ライトするなどし、BACnet システムを構築します。

(2) 用語について

用語	意味
AnnexU	BACnet/IPv6 を規定する BACnet 規格の追加仕様のこと。 詳細は BACnet/IPv6 を参照。
ArrayIndex	配列構造となっているプロパティに使われるもので配列の番号を識別するもの。
B-BC	BACnet Building Controller の略。 BACnet デバイスのことであり、一般的な目的で、すべてのビルオートメーション機能と制御機能を実行することができる BACnet 機器のこと。
B-OWS	BACnet Operator Workstation の略。 BACnet システム中にあるオペレータインタフェースのこと。
BACnet/IP	IPv4 アドレスを使用する通信に対応した BACnet の標準仕様のこと。 日本国内ではこの標準仕様で IPv6 アドレスを使用できるように動作が拡張されることもあるが、BACnet 規格としては非標準である。
BACnet/IPv6	IPv6 アドレスを使用する通信に対応した BACnet の標準仕様のこと。 BACnet/IP との互換性は無く、BACnet/IPv6 同士で通信する必要がある。
BACnet オブジェクト (Object)	BACnet 対応装置が持っている入出力や内部状態に関する情報のこと。オブジェクトは各種プロパティによって構成されている。BACnet 規格には多くのオブジェクトタイプが用意されている。
BACnet サービス	BACnet 対応装置が外部の BACnet 対応装置との間で提供し合う機能のこと。
BACnet デバイス	BACnet システムに接続される機器のこと。 狭義的には、BACnet コントローラ側である BACnet システムの入出力を提供する機器のみを表現することがある。
COV (シーオーブイ)	Change of Value の略。「規定値変化」と訳される。BACnet 規格で定められている通信仕様の一つであり PresentValue または StatusFlags の値が変化した時に発信されるイベントのこと。
Event (イベント)	BACnet 規格で定められている通信仕様のこと。
EventState プロパティ (イベント状態)	BACnet プロパティの 1 つ。「イベント状態」と訳される。多くのオブジェクトが持っているプロパティであり、Reliability プロパティなどと連動し、イベントの状態を管理している。
HIM (ヒム)	Human Interface Module の略。日本電気設備学会特有の用語。 BACnet オペレータワークステーションのことであり、BACnet システムの上位側として、BACnet システムの状態を管理、監視するための中央監視装置のこと。
ICONT (Icont) (アイコント)	Intelligent Controller の略。日本電気設備学会特有の用語。 BACnet コントローラのことであり、BACnet システムの下位側として、BACnet システムの入出力を提供する機器のこと。
Out_Of_Service (アウトオブサービス)	BACnet プロパティの 1 つ。「サービス外」と訳される。多くのオブジェクトが持っているプロパティであり、True 状態にある時に、入出力デバイスが物理的に切り離されていることを意味する。
Present_Value	BACnet プロパティの 1 つ。「現在値」と訳される。多くのオブジェクトが持っているプロパティであり、オブジェクトに関連付けられた入出力値などを取り扱う中心的なプロパティである。

用語	意味
Priority (プライオリティ)	「優先度」と略される。通常は、プロパティのライト要求を行う際の優先度を示し、1～16 までである。ライト要求は、同じか、より小さい値の Priority 値でしか受け付けられない。
Reliability (リライアビリティ)	BACnet プロパティの 1 つ。「信頼性」と訳される。多くのオブジェクトが持っているプロパティであり、入出力デバイスが正常状態であるのか異常状態にあるのかを管理する。
インスタンス番号 (Instance No)	BACnet デバイス毎や BACnet オブジェクトタイプ毎には番号が割り付けられており、BACnet システムではこの番号によって各種 BACnet サービスを提供し合う。この番号のことをインスタンスと称し、本書では「インスタンス番号」または「InstanceNo」と記述する。
オブジェクト ID (ObjectID) (ObjectIdentifier)	オブジェクトタイプと、そのオブジェクトのインスタンス番号からなるオブジェクトの識別子のこと。
オブジェクトタイプ (Object Type)	BACnet には、BI/BO/BV や AI/AO/AV といったさまざまな目的別に用意されたオブジェクトと称するものでシステムの状態を管理または制御する。オブジェクトタイプとは、これらオブジェクトの種類のこと。
オプション	実装するかどうかは、BACnet システム毎に決めればよいとされている部分のこと。BACnet 仕様には、プロパティの有無など多くの点で、オプション機能が数多く存在している。
再初期化	一度 BACnet システムから離脱し、BACnet コントローラの再起動または情報を初期化するなどして、再度 BACnet へ参入し直すこと。
参入	他の BACnet デバイスに対し、自分の存在を通知し、BACnet システムに入ること。 参入処理の方法や手順は、各 BACnet 規格により異なっている。
デバイスインスタンス番号 (Device Instance No)	BACnet デバイス毎に割り付けられた番号のこと。 デバイスインスタンス番号により、接続機器を識別する。
プロパティ (Property)	各オブジェクトの詳細情報、属性のこと。オブジェクトは数多くのプロパティから構成されており、プロパティの状態によって、他のプロパティへ影響を与え制御や変化することがある。
プロパティ ID (PropertyID) (PropertyIdentifier)	プロパティの種類を区別する識別子のこと。
離脱	BACnet システムから抜けること。

(3) BACnet デバイスのプロフィール

名称	略称
BACnet Advanced Operator Workstation	B-AWS
BACnet Operator Workstation	B-OWS
BACnet Operator Display	B-OD
BACnet Building Controller	B-BC
BACnet Advanced Application Controller	B-AAC
BACnet Application Specific Controller	B-ASC
BACnet Smart Actuator	B-SA
BACnet Smart Sensor	B-SS

ポイント プロファイルと同じ用語の使い方

BACnet デバイスを表現する際、プロファイルと同じ名称を用いることがあります。一般的な機能要件を満足するものを意図してこれらの用語を使用する 경우가多く、プロファイルが要求する機能要件をすべて満たすことを必ずしも意図しないことにご注意ください。

1. 8. ご提供の方法とサポート

(1) 提供形態

項目	内容
DLL 開発用キット	DLL プログラム lib ファイルおよび DriverType. h マニュアル(本書) サンプルソース 開発用ライセンス (無制限) 1 年間のテクニカルサポート
運用ライセンス	運用ライセンス

- 運用用ライセンスは DLL 開発用キットをお求めになった場合のみのご提供となります。
- DLL 開発用キットおよび年間保守は、1 事業所内、1 プロジェクトの窓口単位でご契約いただきます。
- 有償で、別途年間保守をつけることができます。

(2) 開発キットについて

- DLL 開発用キットご購入時、別途使用許諾契約書(※)への同意が必要となります。
- DLL 開発用キットは、1 事業所内、1 プロジェクト毎での販売となります。

(※)使用範囲内など、詳細は別途定める使用許諾契約書に従います。

(3) 運用ライセンスについて

- 運用時には DLL が搭載された 1 PC/ 1 OS につき、1 つの運用ライセンスが必要です。

(※)使用範囲内など、詳細は別途定める使用許諾契約書に従います。

(4) テクニカルサポートについて

- 本製品をご利用した開発の、本製品に関する開発サポートを行います。
- 2 年目以降のご相談には年間保守費が必要です。
- お問い合わせは、原則として E メールでお願いいたします。
- 技術者派遣のご要望がありましたら、有償にて対応させていただきます。
- 緊急対応には応じることができない場合がありますので、ご了承ください。
- 上記以外については、別費用となりますが、ご相談の上、対応させていただきます。

(5) バージョンアップ

- 本製品は、事前のご連絡なしに機能拡張としてバージョンアップが行われます。
- 技術サポート期間中に限り、バージョンアップ情報をお届けいたします。バージョンアップは、内容と規模により、有償となる場合があります。
- 技術サポート終了時にバージョンアップ版を必要とされる場合は、改めて初期費用が発生いたします。

※本書末の「5. 2 保証について」「5. 3. サポートについて」もあわせて参照ください。

第2章 仕様

2.1. BACnet 規格

本製品が準拠できる BACnet の規格は次の通りです。

- ANSI/ASHRAE Standard 135-2004
- ANSI/ASHRAE Standard 135-2010
- ANSI/ASHRAE Standard 135-2012
- IEC 61850-3:2004
- IEC 61850-3:2004 アドendum a
- IEC 61850-3:2006 アドendum a
- IEC 61850-3:2017

2.2. BACnet 通信

接続できる BACnet デバイスに関する仕様は次の通りです。

オブジェクトを作成するとパソコンのメモリを使用しますので、使用されるメモリ量に注意してください。

BACnet デバイス数	最大推奨 200 台
管理できるオブジェクト数	最大推奨 10,000 点

2.3. サポートオブジェクト

本製品が対応する規格、オブジェクトは、次の通りです。設定した規格で対応していないと表記しているオブジェクトでも追加すること自体は可能となっています。

本ユニットが実装するオブジェクト			規格		
タイプ 番号	名称	略号	ANSI/ASHRAE Standard 135-2004 (ISO2003) 135-2010 135-2012	IEIEJ-P-0003:2000 IEIEJ-P-0003:2000 IEIEJ-P-0003:2000 IEIEJ-P-0003:2000	IEIEJ-G-0006:2006 IEIEJ-G-0006:2006 IEIEJ-G-0006:2006 IEIEJ-G-0006:2006
0	Analog Input Object Type	AI	○	○	○
1	Analog Output Object Type	AO	○	○	○
2	Analog Value Object Type	AV	○	○	○
3	Binary Input Object Type	BI	○	○	○
4	Binary Output Object Type	BO	○	○	○
5	Binary Value Object Type	BV	○	○	○
13	Multi-state Input Object Type	MI	○	○	○
14	Multi-state Output Object Type	MO	○	○	○
19	Multi-state Value Object Type	MV	○	○	○
23	Accumulator Object Type	AC	○	—	○
6	Calendar Object Type	CA	○	○	○
11	Group Object Type	GR	○	○	○
15	Notification Class Object Type	NC	○	○	○
17	Schedule Object Type	SC	○	○	○
20	TrendLog Object Type	TL	○	○	○
128	計量オブジェクト	計量	—	○	—
130	電力デマンド監視オブジェクト	電力監視	—	○	○
131	電力デマンド制御オブジェクト	電力制御	—	○	○
132	発電機負荷制御オブジェクト	発電機	—	○	○
8	Device Object Type	DV	○	○	○

2. 4. サポート BIBB

本製品が実装する BIBB は、次の通りです（○はサポート、×は未サポートを意味します）。
BIBB サポート状況は BACnet2012 規格の定義を基準としています。

(1) Data Sharing BIBBs

項目	略称
ReadProperty-A	DS-RP-A
ReadProperty-B	DS-RP-B
ReadPropertyMultiple-A	DS-RPM-A
ReadPropertyMultiple-B	DS-RPM-B
WriteProperty-A	DS-WP-A
WriteProperty-B	DS-WP-B
WritePropertyMultiple-A	DS-WPM-A
WritePropertyMultiple-B	DS-WPM-B
COV-A	DS-COV-A
COV-B	DS-COV-B
COV-Unsubscribed-A	DS-COVU-A
COV-Unsubscribed-B	DS-COVU-B

(2) Alarm and Event Management BIBBs

項目	略称
Notification-A *1	AE-N-A
Notification Internal-B	AE-N-I-B

*1 サポートするイベントタイプは以下のとおりです。

番号	イベントタイプ
1	ChangeOfState
2	ChangeOfValue
3	CommandFailure
4	FloatingLimit
5	OutOfRange
10	BufferReady
11	UnsignedRange
19	ChangeOfReliability

(3) Scheduling BIBBs

項目	略称
Internal-B	SCHED-I-B
External-B	SCHED-E-B
Weekly Schedule-A	SCHED-WS-A
Weekly Schedule Internal-B	SCHED-WS-I-B

(4) Trending BIBBs

項目	略称
Viewing and Modifying Trends Internal-B	T-VMT-I-B
Viewing and Modifying Trends External-B	T-VMT-E-B
Automated Trend Retrieval-B	T-ATR-B

(5) Device and Network Management BIBBs

項目	略称
Dynamic Device Binding-A	DM-DDB-A
Dynamic Device Binding-B	DM-DDB-B
Dynamic Object Binding-B	DM-DOB-B
DeviceCommunicationControl-A	DM-DCC-A
TimeSynchronization-A	DM-TS-A
TimeSynchronization-B	DM-TS-B
UTCTimeSynchronization-A	DM-UTC-A
UTCTimeSynchronization-B	DM-UTC-B
ReinitializeDevice-A	DM-RD-A
Restart-A	DM-R-A
Restart-B	DM-R-B
List Manipulation-A	DM-LM-A
List Manipulation-B	DM-LM-B
Automatic Time Synchronization-A	DM-ATS-A
Manual Time Synchronization-A	DM-MTS-A

2.5. オブジェクトプロパティ

本製品で実装できるオブジェクトプロパティは次の通りです。

CharacterString 型のプロパティは 511 バイトまでの文字列を保持することができます。

ID が 9000 番台のプロパティは本製品特有のプロパティです。詳細は「2.6. オプションプロパティ」を参照ください。

【次ページの表について】

- 「適合」欄は、BACnet の仕様として定められたプロパティの存在性を示します。

O	オプションのプロパティであることを示す。
R	必須プロパティであり、BACnet サービスを用いてリードが可能であることを示す。
W	必須プロパティであり、BACnet サービスを用いてリードとライトが可能であることを示す。
EX	本ユニット特有のプロパティであることを示す。

- オプションの使用契機や、プロパティの意味に関しては、BACnet 仕様書を参照してください。
- 「R/W」欄は、本製品の実装状況を示し、外部から本ユニットに対して BACnet サービスを利用してリードまたはライトできるかどうかを示します。

R	外部からリードのみ行うことができる。
W	外部からライトのみ行うことができる。
R/W	外部からリードとライトの両方を行うことができる。

2.5.1. Analog Input Object Type (AI)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
17	Notification_Class	Unsigned	0	R/W
22	COV_Increment	REAL	0	R/W
25	Deadband	REAL	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
45	High_Limit	REAL	0	R/W
52	Limit_Enable	BIT STRING	0	R/W
59	Low_Limit	REAL	0	R/W
65	Max_Pres_Value	REAL	0 *2	R/W
69	Min_Pres_Value	REAL	0 *2	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	REAL	R *1	R/W
103	Reliability	ENUMERATED	0 *1	R/W
106	Resolution	REAL	0	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
117	Units	ENUMERATED	R	R/W
118	Update_Interval	Unsigned	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *5	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *3	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *3	R/W
353	Event_Detection_Enable	BOOLEAN	0 *3	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *3	R/W
356	Time_Delay_Normal	Unsigned	0 *3	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *3	R
9001	力率	BOOLEAN	EX	R/W
9002	Intrinsic_Event_Disable	BOOLEAN	EX *4	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 これらのプロパティを使用する場合、BD_SetPropertyUnused で使用可能にする必要があります。

*3 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*4 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*5 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.2. Analog Output Object Type (AO)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
17	Notification_Class	Unsigned	0	R/W
22	COV_Increment	REAL	0	R/W
25	Deadband	REAL	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
45	High_Limit	REAL	0	R/W
52	Limit_Enable	BIT STRING	0	R/W
59	Low_Limit	REAL	0	R/W
65	Max_Pres_Value	REAL	0 *2	R/W
69	Min_Pres_Value	REAL	0 *2	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	REAL	R	R/W
87	PriorityArray	BACnetPriorityArray	0	R
103	Reliability	ENUMERATED	0 *1	R/W
104	RelinquishDefault	REAL	R	R/W
106	Resolution	REAL	0	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
117	Units	ENUMERATED	R	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *5	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *3	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *3	R/W
353	Event_Detection_Enable	BOOLEAN	0 *3	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *3	R/W
356	Time_Delay_Normal	Unsigned	0 *3	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *3	R
9001	力率	BOOLEAN	EX	R/W
9002	Intrinsic_Event_Disable	BOOLEAN	EX *4	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 これらのプロパティを使用する場合、BD_SetPropertyUnused で使用可能にする必要があります。

*3 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*4 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*5 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.3. Analog Value Object Type (AV)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
17	Notification_Class	Unsigned	0	R/W
22	COV_Increment	REAL	0	R/W
25	Deadband	REAL	0	R/W
28	Description	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
45	High_Limit	REAL	0	R/W
52	Limit_Enable	BIT STRING	0	R/W
59	Low_Limit	REAL	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	REAL	R	R/W
87	PriorityArray	BACnetPriorityArray	0	R
103	Reliability	ENUMERATED	0 *1	R/W
104	RelinquishDefault	REAL	0	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
117	Units	ENUMERATED	R	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9001	力率	BOOLEAN	EX	R/W
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9004	Value_Output	BOOLEAN	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*3 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.4. Binary Input Object Type (BI)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
4	Active_Text	CharacterString	0	R/W
6	Alarm_Value	ENUMERATED	0	R/W
15	Change_Of_State_Count	Unsigned	0	R/W
16	Change_Of_State_Time	BACnetDateTime	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
33	Elapsed_Active_Time	Unsigned32	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
46	Inactive_Text	CharacterString	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
84	Polarity	ENUMERATED	R	R/W
85	Present_Value	ENUMERATED	R *1	R/W
103	Reliability	ENUMERATED	0 *1	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
114	Time_Of_Active_Time_Reset	BACnetDateTime	0	R/W
115	Time_Of_State_Count_Reset	BACnetDateTime	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*3 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.5. Binary Output Object Type (BO)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
4	Active_Text	CharacterString	0	R/W
15	Change_Of_State_Count	Unsigned	0	R/W
16	Change_Of_State_Time	BACnetDateTime	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
33	Elapsed_Active_Time	Unsigned32	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
40	Feedback_Value	ENUMERATED	0	R
46	Inactive_Text	CharacterString	0	R/W
66	Minimum_Off_Time	Unsigned32	0	R/W
67	Minimum_On_Time	Unsigned32	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
84	Polarity	ENUMERATED	R	R/W
85	Present_Value	ENUMERATED	R	R/W
87	PriorityArray	BACnetPriorityArray	0	R
103	Reliability	ENUMERATED	0 *1	R/W
104	RelinquishDefault	ENUMERATED	R	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
114	Time_Of_Active_Time_Reset	BACnetDateTime	0	R/W
115	Time_Of_State_Count_Reset	BACnetDateTime	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*3 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.6. Binary Value Object Type (BV)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
4	Active_Text	CharacterString	0	R/W
6	Alarm_Value	ENUMERATED	0	R/W
15	Change_Of_State_Count	Unsigned32	0	R/W
16	Change_Of_State_Time	BACnetDateTime	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
33	Elapsed_Active_Time	Unsigned32	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
46	Inactive_Text	CharacterString	0	R/W
66	Minimum_Off_Time	Unsigned32	0	R/W
67	Minimum_On_Time	Unsigned32	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	ENUMERATED	R	R/W
87	Priority_Array	BACnetPriorityArray	0	R
103	Reliability	ENUMERATED	0 *1	R/W
104	RelinquishDefault	ENUMERATED	0	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
114	Time_Of_Active_Time_Reset	BACnetDateTime	0	R/W
115	Time_Of_State_Count_Reset	BACnetDateTime	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9004	Value_Output	BOOLEAN	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

- *2 BACnet 規格が BACnet2012 以降の場合のみ有効です。
- *3 BACnet 規格が BACnet2010 以前の場合のみ有効です。
- *4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.7. Multi-state Input Object Type (MI)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
7	Alarm_Values	BACnetLIST of Unsigned	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
39	Fault_Values	BACnetLIST of Unsigned	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
74	Number_Of_States	Unsigned	R	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned	R *1	R/W
103	Reliability	ENUMERATED	0 *1	R/W
110	State_Text	BACnetARRAY[N] of CharacterString	0	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

- *1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。
- *2 BACnet 規格が BACnet2012 以降の場合のみ有効です。
- *3 BACnet 規格が BACnet2010 以前の場合のみ有効です。
- *4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.8. Multi-state Output Object Type (MO)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
40	Feedback_Value	Unsigned	0	R
72	Notify_Type	ENUMERATED	0	R/W
74	Number_Of_States	Unsigned	R	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned	R	R/W
87	PriorityArray	BACnetPriorityArray	0	R
103	Reliability	ENUMERATED	0 *1	R/W
104	RelinquishDefault	Unsigned	R	R/W
110	State_Text	BACnetARRAY[N] of CharacterString	0	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*3 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.9. Multi-state Value Object Type (MV)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
7	Alarm_Values	BACnetLIST of Unsigned	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
39	Fault_Values	BACnetLIST of Unsigned	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
74	Number_Of_States	Unsigned	R	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned	R	R/W
87	PriorityArray	BACnetPriorityArray	0	R
103	Reliability	ENUMERATED	0 *1	R/W
104	Relinquish_Default	Unsigned	0	R/W
110	State_Text	BACnetARRAY[N] of CharacterString	0	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9004	Value_Output	BOOLEAN	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*3 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.10. Accumulator Object Type (AC)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
45	High_limit	Unsigned	0	R/W
52	Limit_Enable	BIT STRING	0	R/W
59	Low_limit	Unsigned	0	R/W
65	Max_Pres_Value	Unsigned	R	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned	R *1	R/W
103	Reliability	ENUMERATED	0 *1	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
117	Units	ENUMERATED	R	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
168	Profile_Name	CharacterString	0	R/W
182	Limit_Monitoring_Interval	Unsigned	0	R/W
183	Logging_Object	BACnetObjectIdentifier	0	R/W
184	Logging_Record	BACnetAccumulatorRecord	0	R
185	Prescale	BACnetPrescale	0	R/W
186	Pulse_Rate	Unsigned	0 *1	R/W
187	Scale	BACnetScale	R	R/W
190	Value_Before_Change	Unsigned	0	R
191	Value_Set	Unsigned	0	R/W
192	Value_Change_Time	BACnetDateTime	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
356	Time_Delay_Normal	Unsigned	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX *3	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*3 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.11. Calendar Object Type (CA)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
23	Date_List	BACnetLIST of BACnetCalendarEntry	R	R/W
28	Description	CharacterString	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
85	Present_Value	BOOLEAN	R	R
168	Profile_Name	CharacterString	0	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *1	R

*1 BACnet 規格が BACnet2012 以降の場合のみ有効です。

2.5.12. Group Object Type (GR)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
28	Description	CharacterString	0	R/W
53	List_Of_Group_Members	BACnetLIST of ReadAccessSpecification	R	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
85	Present_Value	BACnetLIST of ReadAccessResult	R	R
168	Profile_Name	CharacterString	0	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *1	R

*1 BACnet 規格が BACnet2012 以降の場合のみ有効です。

2.5.13. Notification Class Object Type (NC)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
1	Ack_Required	BIT STRING	R	R/W
17	Notification_Class	Unsigned	R	R
28	Description	CharacterString	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
86	Priority	BACnetARRAY[3] of Unsigned	R	R/W
102	Recipient_List	BACnetLIST of BACnetDestination	R	R/W
168	Profile_Name	CharacterString	0	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *1	R

*1 BACnet 規格が BACnet2012 以降の場合のみ有効です。

2.5.14. Schedule Object Type (SC)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
28	Description	CharacterString	0	R/W
32	Effective_Period	BACnetDateRange	R	R/W
38	Exception_Schedule	BACnetARRAY[N] of BACnetSpecialEvent	0	R/W
54	List_Of_Object_Property_References	BACnetLIST of BACnetDeviceObjectPropertyReference	R	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Any	R *1	R/W
88	Priority_For_Writing	Unsigned (1..16)	R	R/W
103	Reliability	ENUMERATED	0	R
111	Status_Flags	BIT STRING	R	R
123	Weekly_Schedule	BACnetARRAY[7] of BACnetDailySchedule	0	R/W
168	Profile_Name	CharacterString	0	R/W
174	Schedule_Default	Any	R	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R
9011	Schedule_Output_Disable	BOOLEAN	EX	R/W
9012	Schedule_Expand	BOOLEAN	EX	R/W
9014	Schedule_Default_Disable	BOOLEAN	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

2.5.15. Trend Log Object Type (TL)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
103	Reliability	ENUMERATED	0 *1	R
111	Status_Flags	BIT STRING	R *1	R
126	Buffer_Size	Unsigned32	R	R/W
127	Client_COV_Increment	BACnetClientCOV	0	R/W
128	COV_Resubscription_Interval	Unsigned	0	R/W
129	Current_Notify_Time	BACnetDateTime	0 *3	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *4	R/W
131	Log_Buffer	BACnetLIST of BACnetLogRecord	R	R
132	Log_DeviceObjectProperty	BACnetDeviceObjectPropertyReference	0	R/W
133	Enable	BOOLEAN	W	R/W
134	Log_Interval	Unsigned	0	R/W
137	Notification_Threshold	Unsigned32	0	R/W
138	Previous_Notify_Time	BACnetDateTime	0 *3	R/W
140	Records_Since_Notification	Unsigned32	0	R/W
141	Record_Count	Unsigned32	W	R/W
142	Start_Time	BACnetDateTime	0	R/W
143	Stop_Time	BACnetDateTime	0	R/W
144	Stop_When_Full	BOOLEAN	R	R/W
145	Total_Record_Count	Unsigned32	R	R/W
168	Profile_Name	CharacterString	0	R/W
173	Last_Notify_Record	Unsigned32	0	R/W
197	Logging_Type	ENUMERATED	R *1	R/W
205	Trigger	BOOLEAN	0 *1	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *2	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *2	R/W
353	Event_Detection_Enable	BOOLEAN	0 *2	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *2	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *2	R

*1 BACnet 規格が BACnet2010 以降の場合のみ有効です。

*2 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*3 BACnet 規格が BACnet2004 以前の場合のみ有効です。

*4 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.16. 計量オブジェクト

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
31	Device_Type	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
52	Limit_Enable	BIT STRING	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned	W *1	R/W
103	Reliability	ENUMERATED	0 *1	R/W
111	Status_Flags	BIT STRING	R	R
117	Units	ENUMERATED	R	R/W
520	最大カウント値	Unsigned	0	R/W
521	重み	REAL	0	R/W
522	リセット初期値	Unsigned	0	R/W
523	リセット直前値	Unsigned	0	R/W
524	リセット時刻	BACnetDateTime	0	R/W
525	上限監視積算時間	Unsigned	R	R/W
526	上限値	Unsigned	0	R/W
527	時系列データリスト	BACnetLIST of HistoricalData	R	R/W
530	時系列データ最新更新時刻	BACnetDateTime	R	R
9002	Intrinsic_Event_Disable	BOOLEAN	EX	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

2.5.17. 電力デマンド監視オブジェクト

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
7	Alarm_Values	BACnetLIST of Unsigned	0	R/W
17	Notification_Class	Unsigned	R	R/W
28	Description	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
39	Fault_Values	BACnetLIST of Unsigned	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned (1..3)	R *2	R/W
103	Reliability	ENUMERATED	0 *1	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
117	Units	ENUMERATED	R	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *5	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *3	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *3	R/W
353	Event_Detection_Enable	BOOLEAN	0 *3	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *3	R/W
356	Time_Delay_Normal	Unsigned	0 *3	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *3	R
560	Start_Time_Of_Monitoring (監視開始時分)	Time	R	R/W
561	Time_Of_Alarm_Lock (警報ロック時間)	Unsigned (0..30)	R	R/W
562	Contract_Receiving_Power (契約電力)	REAL	W	R/W
563	Target_Value_Of_Power (目標電力)	REAL	W	R/W
564	Alarm_Value_Of_Power (警報電力)	REAL	W	R/W
565	Elapsed_Time (経過時間)	Unsigned (1..30)	R	R/W
566	Consumed_WH_In_This_Term (今回周期の使用電力量)	BACnetARRAY[30] of ListOfWHData	R	R/W
567	Estimated_Power (予測電力量)	REAL	R	R/W
568	Adjust_Power (調整電力量)	REAL	R	R/W
569	Time_Width_Of_WH_Trend (傾向時間間隔)	Unsigned (0..5)	R	R/W
580	Consumed_WH_In_Last_Term (前回周期の使用電力量)	BACnetARRAY[30] of ListOfWHData	R	R/W
581	List_Of_Pulse_Counter_Reference	BACnetLIST of BACnetDeviceObjectPropertyReference	0	R/W

9002	Intrinsic_Event_Disable	BOOLEAN	EX *4	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が IEC 61850-3:2000 (アデンダム a も含む) の場合、値の範囲は (0..2) となります。

*3 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*4 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*5 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.18. 電力デマンド制御オブジェクト

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
7	Alarm_Values	BACnetLIST of Unsigned	0	R/W
17	Notification_Class	Unsigned	R	R/W
28	Description	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
39	Fault_Values	BACnetLIST of Unsigned	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned (1..16)	R *2	R/W
88	Property_For_Writing	Unsigned (1..16)	R	R/W
103	Reliability	ENUMERATED	0 *1	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *5	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *3	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *3	R/W
353	Event_Detection_Enable	BOOLEAN	0 *3	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *3	R/W
356	Time_Delay_Normal	Unsigned	0 *3	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *3	R
570	Level_Value(レベル設定電力)	BACnetARRAY[16] of REAL	0	R/W
9002	Intrinsic_Event_Disable	BOOLEAN	EX *4	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が IEC 61850-3:2000 (アデンダム a も含む) の場合、値の範囲は (0..15) となります。

*3 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*4 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*5 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.19. 発電機負荷制御オブジェクト

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
0	Acked_Transitions	BIT STRING	0	R/W
7	Alarm_Values	BACnetLIST of Unsigned	0	R/W
17	Notification_Class	Unsigned	0	R/W
28	Description	CharacterString	0	R/W
35	Event_Enable	BIT STRING	0	R/W
36	Event_State	ENUMERATED	R	R
39	Fault_Values	BACnetLIST of Unsigned	0	R/W
72	Notify_Type	ENUMERATED	0	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
81	Out_Of_Service	BOOLEAN	R	R/W
85	Present_Value	Unsigned (1..16)	R *4	R/W
88	Priority_For_Writing	Unsigned (1..16)	R	R/W
103	Reliability	ENUMERATED	0 *1	R/W
111	Status_Flags	BIT STRING	R	R
113	Time_Delay	Unsigned	0	R/W
130	Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	0 *7	R/W
168	Profile_Name	CharacterString	0	R/W
351	Event_Message_Texts	BACnetARRAY[3] of CharacterString	0 *5	R
352	Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	0 *5	R/W
353	Event_Detection_Enable	BOOLEAN	0 *5	R/W
354	Event_Algorithm_Inhibit	BOOLEAN	0 *5	R/W
356	Time_Delay_Normal	Unsigned	0 *5	R/W
371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *5	R
570	Level_Value(レベル設定電力)	BACnetARRAY[16] of REAL	0 *2	R/W
576	Target_Value_To_Supply (発電機目標電力)	REAL	R	R/W
577	Adjust_Value_Of_Load (発電機調整電力)	REAL	R	R/W
578	発電機レベル設定電力	BACnetARRAY[16] of REAL	W *3	R/W
9002	Intrinsic_Event_Disable	BOOLEAN	EX *6	R/W
9003	Unsolicited_COV	ENUMERATED	EX	R/W
9006	COV_Send_Interval	Unsigned	EX	R/W
9027	Unsolicited_COV_Process_ID	Unsigned	EX	R/W

*1 Out_Of_Service が TRUE の場合にのみ書き込み可能です。

*2 BACnet 規格が IEC 61850-3:2000 (アデンダム a も含む) 以外の場合のみ有効です。

*3 BACnet 規格が IEC 61850-3:2000 (アデンダム a も含む) の場合のみ有効です。

*4 BACnet 規格が IEC 61850-3:2000 (アデンダム a も含む) の場合、値の範囲は (0..15) となります。

*5 BACnet 規格が BACnet2012 以降の場合のみ有効です。

*6 BACnet 規格が BACnet2010 以前の場合のみ有効です。

*7 BD_SetProperty 等の関数での書き込みを行うことができません。

2.5.20. Device Object Type (DV)

ID	PROPERTY IDENTIFIER	APPLICATION DATA TYPE	適合	R/W
5	Active_VT_Sessions	BACnetLIST of BACnetVTSession	0 *5	R
10	APDU_Segment_Timeout	Unsigned	0	R/W
11	APDU_Timeout	Unsigned	R	R/W
12	Application_Software_Version	CharacterString	R	R/W
24	Daylight_Savings_Status	BOOLEAN	0	R/W
28	Description	CharacterString	0	R/W
30	Device_Address_Binding	BACnetLIST of BACnetAddressBinding	R	R
44	Firmware_Revision	CharacterString	R *1	R
55	List_Of_Session_Keys	BACnetLIST of BACnetSessionKey	0 *5	R
56	Local_Date	Date	0	R
57	Local_Time	Time	0	R
58	Location	CharacterString	0	R/W
62	Max_APDU_Length_Accepted	Unsigned	R	R/W
63	Max_Info_Frames	Unsigned	0	R/W
64	Max_Master	Unsigned (1..127)	0	R/W
70	Model_Name	CharacterString	R	R/W
73	Number_Of_APDU_Retries	Unsigned	R	R/W
75	Object_Identifier	BACnetObjectIdentifier	R	R
76	Object_List	ARRAY[N] of BACnetObjectIdentifier	R	R
77	Object_Name	CharacterString	R	R/W
79	Object_Type	ENUMERATED	R	R
96	Protocol_Object_Types_Supported	BIT STRING	R *1	R
97	Protocol_Services_Supported	BIT STRING	R *1	R
98	Protocol_Version	Unsigned	R	R
107	Segmentation_Supported	ENUMERATED	R	R/W
112	System_Status	ENUMERATED	R	R/W
116	Time_Synchronization_Recipients	BACnetLIST of BACnetRecipient	0	R/W
119	UTC_Offset	INTEGER	0	R/W
120	Vendor_Identifier	Unsigned16	R	R
121	Vendor_Name	CharacterString	R	R
122	VT_Classes_Supported	BACnetLIST of BACnetVTClass	0 *5	R
139	Protocol_Revision	Unsigned	R	R/W
152	Active_COV_Subscriptions	BACnetLIST of BACnetCOVSubscriptions	0	R
155	Database_Revision	Unsigned	R	R/W
167	Max_Segments_Accepted	Unsigned	0	R/W
168	Profile_Name	CharacterString	0	R/W
193	Align_Intervals	BOOLEAN	0 *3	R/W
195	Interval_Offset	Unsigned	0 *3	R/W
196	Last_Restore_Reason	ENUMERATED	0	R/W
202	Restart_Notification_Recipients	BACnetLIST of BACnetRecipient	0	R/W
203	Time_of_Device_Restart	BACnetTimeStamp	0	R/W
204	Time_Synchronization_Interval	Unsigned	0 *3	R/W
206	UTC_Time_Synchronization_Recipients	BACnetLIST of BACnetRecipient	0	R/W

371	Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R *4	R
-----	---------------	--	------	---

- *1 BD_SetProperty 等の関数での書き込みを行うことができません。
- *2 BACnet 規格が IEIEJ-P-0003:2000（アデンダム a も含む）以外の場合のみ有効です。
- *3 BACnet 規格が BACnet2010 以降の場合のみ有効です。
- *4 BACnet 規格が BACnet2012 以降の場合のみ有効です。
- *5 BACnet 規格が BACnet2004 以前の場合のみ有効です。

2.6. オプションプロパティ

本製品が実装するオブジェクトには、次のプロパティを実装しているものがあります。

ID	PROPERTY IDENTIFIER	内容								
9001	力率	PresentValue を力率指定します。								
		<table><tr><th>値</th><th>内容</th></tr><tr><td>False</td><td>PresentValue の値を力率としません。</td></tr><tr><td>True</td><td>力率とし、PresentValue の値は -100.0～+100.0 で管理されます。</td></tr></table>	値	内容	False	PresentValue の値を力率としません。	True	力率とし、PresentValue の値は -100.0～+100.0 で管理されます。		
		値	内容							
		False	PresentValue の値を力率としません。							
True	力率とし、PresentValue の値は -100.0～+100.0 で管理されます。									
9002	Intrinsic_Event_Disable	Event 送信の有無を指定します。								
		<table><tr><th>値</th><th>内容</th></tr><tr><td>False</td><td>Event 送信を行います。</td></tr><tr><td>True</td><td>Event 送信を行います。</td></tr></table>	値	内容	False	Event 送信を行います。	True	Event 送信を行います。		
		値	内容							
		False	Event 送信を行います。							
True	Event 送信を行います。									
9003	Unsolicited_COV	PresentValue の変化に対し COV を送信するかどうかを指定します。								
		<table><tr><th>値</th><th>内容</th></tr><tr><td>0</td><td>送信しません。</td></tr><tr><td>1</td><td>変化時のみ送信します。</td></tr><tr><td>2</td><td>変化時と COV_Send_Interval で指定した周期ごとで送信します。</td></tr></table>	値	内容	0	送信しません。	1	変化時のみ送信します。	2	変化時と COV_Send_Interval で指定した周期ごとで送信します。
		値	内容							
		0	送信しません。							
		1	変化時のみ送信します。							
2	変化時と COV_Send_Interval で指定した周期ごとで送信します。									
9004	Value_Output	入出力方向の指定を行います。								
		<table><tr><th>値</th><th>内容</th></tr><tr><td>False</td><td>Value を入力として取り扱います。</td></tr><tr><td>True</td><td>Value を出力として取り扱います。</td></tr></table>	値	内容	False	Value を入力として取り扱います。	True	Value を出力として取り扱います。		
		値	内容							
		False	Value を入力として取り扱います。							
True	Value を出力として取り扱います。									
9006	COV_Send_Interval	Unsolicited_COV で周期送信を指定した際の周期時間を指定します。単位は秒。								
9011	Schedule_Output_Disable	スケジュールによる出力の指定を行います。								
		<table><tr><th>値</th><th>内容</th></tr><tr><td>False</td><td>スケジュールによる出力を行います。</td></tr><tr><td>True</td><td>スケジュールによる出力を禁止します。</td></tr></table>	値	内容	False	スケジュールによる出力を行います。	True	スケジュールによる出力を禁止します。		
		値	内容							
		False	スケジュールによる出力を行います。							
True	スケジュールによる出力を禁止します。									
9012	Schedule_Expand	スケジュールの日替わり時の処理を指定します。								
		<table><tr><th>値</th><th>内容</th></tr><tr><td>False</td><td>7 日目のスケジュールを日替わり時に自動生成しません。</td></tr><tr><td>True</td><td>7 日目のスケジュールを日替わり時に自動生成を行います。</td></tr></table>	値	内容	False	7 日目のスケジュールを日替わり時に自動生成しません。	True	7 日目のスケジュールを日替わり時に自動生成を行います。		
		値	内容							
		False	7 日目のスケジュールを日替わり時に自動生成しません。							
True	7 日目のスケジュールを日替わり時に自動生成を行います。									
9014	Schedule_Default_Disable	Schedule_Default プロパティを無効にします。								
		<table><tr><th>値</th><th>内容</th></tr><tr><td>False</td><td>Schedule_Default を使用します。</td></tr><tr><td>True</td><td>Schedule_Default を使用しません。</td></tr></table>	値	内容	False	Schedule_Default を使用します。	True	Schedule_Default を使用しません。		
		値	内容							
		False	Schedule_Default を使用します。							
True	Schedule_Default を使用しません。									
9027	Unsolicited_COV_Process_ID	Unsolicited_COV で送信する COV の ProcessID を指定します。								

2.7. SystemStatus プロパティ

本製品が扱う Device オブジェクトの System_Status の値は以下の通りです。

番号	ステータス名	内容
0	Operational	運転しており、ネットワークに参加している状態です。
1	Operational ReadOnly	本デバイスに対する読み出しに対してのみ、応答可能な状態です。
2	Download Required	参入処理等でダウンロードを必要としている状態です。
3	Download In Progress	参入処理等でダウンロードが実行中の状態です。
4	Non Operational	運転しておらず、ネットワークに参加していない状態です。
5	Backup In Progress	バックアップが実行中の状態です。
6	Unknown	不明な状態です。

第3章 機能

以下に、本製品の機能について記載します。

3.1. オブジェクト作成機能

自身が持つオブジェクトを自由に作成することができます。本ドライバは他の **BACnet** デバイスからリードやライトなどのアクセス要求があると、自動的に応答を返します。

この時、他の **BACnet** デバイスからライト要求があるとコールバック関数で通知されます。

3.2. プロパティアクセス機能

他の **BACnet** デバイス内の **BACnet** オブジェクトのプロパティをアクセスするための関数が用意されています。プロパティへのアクセスは、**ReadProperty**, **ReadPropertyMultiple**, **WriteProperty**, **WritePropertyMultiple**, **AddListElement**, **RemoveListElement**, **ReadRange** サービスなどを使用します。

単純なデータ型(論理、整数、実数、列挙、32Bit までの **BitString**、文字列)であれば、**BACnet** オブジェクトの種別にかかわらずアクセスすることができます。また、単純なデータ型以外では **BACnet** オブジェクトの種別毎にアクセス関数が用意されています。

アクセスする他の **BACnet** デバイスはデバイス管理に登録されている必要があります。登録されていない場合は **NonOperational** と判断されサービスの送信が行われない場合があります。

3.3. 定期収集機能

外部の **BACnet** デバイス内にある **BACnet** オブジェクトのプロパティを指定した周期でリードします。リードした情報は、アプリケーション側より任意のタイミングで読み込むことができます。またリードしたタイミングでコールバックすることもできます。

収集対象の **BACnet** プロパティは、グループ分けして登録することができ、グループ単位で収集動作の開始、一時停止を制御することができます。

3.4. BACnet デバイス管理機能

自身がアクセスする可能性のある **BACnet** デバイスを登録することにより、**Who-Is** 送信や **I-Am** 受信で IP アドレスを動的に取得することができます。

3.5. コールバック関数機能

各種機能では、リアルタイムに処理が行えるようにコールバック関数を登録し、使用することができます。コールバックの登録関数としては、以下のものが用意されています。

設定関数	内容								
BD_SetDeviceStatusChangeCallback	BACnet デバイスの System_Status プロパティが変化した際に、コールされる関数を登録します。								
BD_SetAccessCallback	<p>コールバックモードが変数に用意されている関数で使用するコールバック関数を設定します。</p> <p>コールバックモードの変数値により本ドライバは下記のような動作を行います。</p> <p>コールバック関数では、必ずハンドルを解放してください。</p> <table><tr><th>値</th><th>内容</th></tr><tr><td>0</td><td>関数に必要な通信処理の完了を待たず呼び出し元に直ちに返る。また通信処理の完了時にもコールバックはしない。</td></tr><tr><td>1</td><td>関数に必要な通信処理の完了を待たず呼び出し元に直ちに返る。通信処理の完了後コールバックする。</td></tr><tr><td>2</td><td>関数に必要な通信処理の完了を待って呼び出し元に返る。この関数を呼んだアプリは最大リトライオーバーするまでブロックされる。</td></tr></table>	値	内容	0	関数に必要な通信処理の完了を待たず呼び出し元に直ちに返る。また通信処理の完了時にもコールバックはしない。	1	関数に必要な通信処理の完了を待たず呼び出し元に直ちに返る。通信処理の完了後コールバックする。	2	関数に必要な通信処理の完了を待って呼び出し元に返る。この関数を呼んだアプリは最大リトライオーバーするまでブロックされる。
値	内容								
0	関数に必要な通信処理の完了を待たず呼び出し元に直ちに返る。また通信処理の完了時にもコールバックはしない。								
1	関数に必要な通信処理の完了を待たず呼び出し元に直ちに返る。通信処理の完了後コールバックする。								
2	関数に必要な通信処理の完了を待って呼び出し元に返る。この関数を呼んだアプリは最大リトライオーバーするまでブロックされる。								
BD_SetScanCallback	定期収集機能により、プロパティデータの読み込みが完了するとコールされる関数を登録します。								
BD_SetScanPropertyCallback	定期収集機能により、プロパティデータの読み込みが完了すると Property 毎にコールされる関数を登録します。								
BD_SetCOVCallback	COV を受信したときにコールされる関数を登録します。								
BD_SetEventCallback	Event を受信したときにコールされる関数を登録します。								
BD_PropertyChangeCallback	自身の BACnet オブジェクトプロパティの値が変更したときにコールされる関数を登録します。								

3.6. 関数一覧

種類	関数名	備考
オープンクローズ	BD_Open	
	BD_OpenDirect	
	BD_OpenDirectB	
	BD_Close	
初期設定	BD_AddMonitorDevice	
	BD_AddRestartNotificationDevice	
	BD_AddRestartNotificationAddress	
	BD_AddScan	
	BD_AddScanGroup	
	BD_AddRemoteDevice	
コールバック設定	BD_SetDeviceStatusChangeCallback	
	BD_SetAccessCallback	
	BD_SetScanCallback	
	BD_SetScanPropertyCallback	
	BD_SetCOVCallback	
	BD_SetEventCallback	
	BD_PropertyChangeCallback	
デバイス管理	BD_GetDeviceStatus	
	BD_GetDeviceAddressBinding	
	BD_SendWhois	
	BD_SendWhoisWithRange	
	BD_SendIam	
	BD_GetMyDeviceInstanceNo	
定期収集	BD_StartStopScan	
	BD_ReadScanData	
	BD_ReadScanMultipleData	
自デバイス制御	BD_SetSystemStatus	
	BD_SetMaxOutStandingPDUCount	
	BD_SetWhoIsRangeSendInterval	
通信	BD_SendTimeSync	
	BD_SendTimeSyncUnicast	
	BD_SendUTCTimeSync	
	BD_SetSubscribeCOV	

種類	関数名	備考
WriteProperty(Multiple) 関連	BD_WriteProperty	
	BD_WritePropertyUnsigned	
	BD_WritePropertyEnumerated	
	BD_WritePropertyReal	
	BD_WritePropertyBool	
	BD_WritePropertyCharacterString	
	BD_WritePropertyEventTransitionBits	
	BD_WritePropertyLimitEnable	
	BD_WritePropertyMultiple	
	BD_WritePropertyMultipleA	
ReadProperty(Multiple) 関連	BD_ReadProperty	
	BD_ReadPropertyUnsigned	
	BD_ReadPropertyEnumerated	
	BD_ReadPropertyReal	
	BD_ReadPropertyUnsignedWithStatus	
	BD_ReadPropertyEnumeratedWithStatus	
	BD_ReadPropertyRealWithStatus	
	BD_ReadPropertyBool	
	BD_ReadPropertyCharacterString	
	BD_ReadPropertyEventTransitionBits	
	BD_ReadPropertyLimitEnable	
	BD_ReadPropertyStatusFlags	
	BD_ReadPropertyMultiple	
	BD_ReadPropertyMultipleA	
List 操作	BD_AddListElement	
	BD_RemoveListElement	
Schedule オブジェクト専用	BD_WriteScheduleObjectPropertyList	
	BD_ReadScheduleObjectPropertyList	
	BD_AddScheduleObjectPropertyList	
	BD_RemoveScheduleObjectPropertyList	
	BD_WriteWeeklySchedule	
	BD_ReadWeeklySchedule	
	BD_WriteExceptionSchedule	
	BD_ReadExceptionSchedule	
	BD_WriteExceptionScheduleForSevenDays	
	BD_ReadExceptionScheduleForSevenDays	
	BD_WriteScheduleEffectivePeriod	
	BD_ReadScheduleEffectivePeriod	
Calendar オブジェクト専用	BD_WriteDateList	
	BD_ReadDateList	
	BD_AddDateList	
	BD_RemoveDateList	
TrendLog オブジェクト専用 ReadRange	BD_ReadRangeTrendLogByPosition	
	BD_ReadRangeTrendLogBySequenceNo	
	BD_ReadRangeTrendLogByTime	

種類	関数名	備考
自身のオブジェクト作成	BD_AddObjectID	

種類	関数名	備考
自身のプロパティアクセス	BD_SetProperty	
	BD_GetProperty	
	BD_SetPropertyUnused	
	BD_SetPVAndReliabilityUnsigned	
	BD_SetPVAndReliabilityBinary	
	BD_SetPVAndReliabilityReal	
	BD_SetFeedBackValueAndReliabilityUnsigned	
	BD_SetFeedBackValueAndReliabilityBinary	
	BD_SetReliability	
	BD_AddPulse	
	BD_SetPropertyCharacterString	
	BD_GetPropertyCharacterString	
	BD_SetPropertyAlarmValues	
	BD_GetPropertyAlarmValues	
	BD_SetPropertyFaultValues	
	BD_GetPropertyFaultValues	
	BD_SetPropertyRecipientList	
	BD_SetPropertyLogDeviceObjectPropertyReference	
その他	BD_SaveBACnetObject	
	BD_SaveBACnetObjectAll	
	BD_SaveObjectCSVFile	
	BD_LoadBACnetObject	
	BD_ReinitializeDevice	
	BD_DeviceCommunicationControl	
	BD_ReleaseHandle	

3. 7. 関数仕様

3. 7. 1. 関数共通の仕様

(1) IP アドレス

- 通信時の IP を指定します。
- IPv4 の場合、“192. 168. 0. 1” という形での文字列で指定します。
- IPv6 の場合、“fe80::1:1:0” という形で文字列を指定します。

(2) DeviceInstanceNo

- BACnet デバイスのインスタンス番号を指定します。
- 範囲は 0 から 0x3FFFFE です。

(3) ObjectID

- オブジェクト ID を指定します。
- 上位 10Bit がオブジェクトのタイプ番号、下位 22Bit がインスタンス番号です。

(4) PropertyID

- プロパティの ID を指定します。
- tSimpleValue で表現可能なデータタイプにのみ対応します。

(5) ArrayIndex

- 列挙型のインデックス番号を指定します。
- 使用しない時は-1 を指定します。
- 番号の上限は 2147483646 です。

(6) コールバックモード

- コールバックのモードを指定します。

値	内容
0	通信完了を待たず呼び出し元に直ちに戻る。 コールバックはしない。
1	通信完了を待たず呼び出し元に直ちに戻る。 完了後コールバックする。
2	通信完了を待って呼び出し元に戻る。

- コールバックモードを登録する際の関数の戻り値は次の通りです。

値	内容
0	コールバックモード=2 が指定されている場合は通信処理が終了したことを意味する。正しく終了したか否かは Result で判断する必要がある。 コールバックモード=0 が指定されている場合は通信処理の起動ができたことを意味する。
1 以上	これはコールバックモードが 1 の場合にのみ戻されるハンドルである。コールバック関数の変数で渡される。このハンドルは BD_ReleaseHandle により明示的に解放しなければならない。
-1 以下	エラー 戻り値の詳細は「(8) リターンコード一覧」を参照。

(7) ObjectHandle

- BD_AddObjectID() 関数の戻り値を指定します。
- ObjectHandle は必ず BD_AddObjectID() 関数の戻り値としてください。

(8) リターンコード一覧

関数コール時に戻されるリターンコードは以下の通りです。

値	名称	詳細
-1	BD_ERROR_ALREADY_OPENED	すでに Open された状態で Open 関数をコールした。
-2	BD_ERROR_ALREADY_CLOSED	すでに Close された状態で Close 関数をコールした。
-3	BD_ERROR_FILE_DATA	バックアップファイルの読み書きに失敗した。
-4	BD_ERROR_ALREADY_EXISTED	追加しようとした情報がすでに登録済みであった。
-5	BD_ERROR_INVALIDDATA	指定した変数の値が不正であった。
-6	BD_ERROR_NODATA	指定した変数の値に関する情報が見つからなかった。
-7	BD_ERROR_DATALINK_OPEN	BACnet.ini に設定された IP アドレスが PC に登録されていないまたは既に使用されているなどの理由により、BACnet 通信の開始に失敗した。
-8	BD_ERROR_NO_RESOURCE	リソース不足により実行できなかった。
-9	BD_ERROR_NOT_OPEN	BACnet 通信が開始されていない。
-10	BD_ERROR_FILE_NOT_FOUND	BD_Open、BD_OpenDirectB を実行した時、FileName および FilePath が指すファイルが存在しないなどの問題が発生した。
-999	BD_ERROR_KEY	プロテクト解除のための解除キーが見つからなかった。 ※プロテクトがかかった DLL を使用している場合のみ。
-9999	BD_ERROR_OTHER	不明のエラーが発生した。

(9) 終了コード

引数でコールバックモードを指定する関数は、tReturnCode 型の変数で実行結果を終了コードとして返します。関数が実行されると同構造体の Result メンバが通信処理中を意味する-1 に設定されます。

コールバックモードが非同期（0 または 1）の場合、バックグラウンド動作が終了すると Result メンバ等が結果に応じたデータに書き換えられます。

3. 7. 2. BD_Open

概 要	BACnet 通信の開始(ini ファイル指定版) [オープンクローズ]	
機 能	指定された「構成ファイル」から情報を取り込み、指定された内容で通信動作が開始できる状態にします。 Device オブジェクトの SystemStatus プロパティは NonOperational となります。	
書 式	int BD_Open(char* FileName);	
引 数	char* FileName	「構成ファイル」の場所を完全パス名で指定します。
戻り値	0	正常にオープンできた。
	0 以外	オープンに失敗した。 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	本ドライバを使用する場合は、最初にこの関数または BD_OpenDirect、BD_OpenDirectB のいずれかを実行しなければいけません。 構成ファイルは ini ファイル形式として、セクション名とキー名を以下に示します。	

(1) 構成ファイル

本ドライバの各種パラメータを設定します。

セクション名	キー名	内容	デフォルト値
BACnet	IPAddress	自身の IPv4/v6 アドレスを設定します。	192. 168. 1. 1
	BroadCastAddress	IPv4 のブロードキャストアドレスまたは IPv6 のマルチキャストアドレスを設定します。	192. 168. 1. 255
	PortNo	ポート番号を設定します。	47808
	NetworkNo	ネットワーク番号を設定します。	0
	DeviceInstanceNo	自身の DeviceInstance 番号を設定します。	254
	AnnexUEnable	BACnet/IPv6 (AnnexU) の有効無効を設定します。 IPAddress キーが IPv6 アドレスかつ、IEIESupportMode キーが ANSI/ASHRAE Standard 135-2012 または IEC 61850-3:2017 の場合のみ使用できます。 0:無効 1:有効	0
	RetryCount	自身が送信した確認付リクエストに回答がなかった場合に、当該リクエストを再送する回数を設定します。	3
	APDUTimeOut	自身が送信した確認付リクエストに対して、「応答がない」と判断し、再送を行うまでの時間をミリ秒単位で設定します。	6000
	MaxOutStandingPDUCount	自身が送信した確認付リクエストに対して応答がないまま連続で送信できるリクエストの最大数を設定します。	20
	SendInterval	自身が連続でパケットを送信する際の送信間隔をミリ秒単位で設定します。(0~1000)	0
	SendIamIntervalSecond	定期的に I-Am を送信する周期(1~600) [秒] を設定します。ただし 0 が指定されると定期的送信は行いません。	60
	MaxScanPropertyCount	定周期スキャン機能を使用した場合の、1 パケットで取りに行くプロパティの数を設定します。	80
	WhoisSendInterval	定期的に監視対象に対して Who-Is を送信する周期。(1~600) [秒] を設定します。 ただし 0 が指定されると定期的送信は行いません。	60
	WhoisTimeOut	デバイス動作監視元から Who-Is が送信されているかを監視する時間。(1~600) [秒] を設定します。 ただし 0 が指定されると監視は行いません。	150

セクション名	キー名	内容	デフォルト値
BACnet	WhoisInitiatorMode	Who-Is の送信モードを設定します。 動作は IEIESupportMode によって異なります。 0:WhoisSendInterval 間隔で常に Who-Is を同報する。 1:バックアップ時に送信する。 2:バックアップのバックアップ時に送信する。 3:Who-Is を送信しない。 ※1, 2 の設定は IEIESupportMode が 2, 6 の時のみ有効 ※0 以外に設定した場合でも BACnet デバイス管理機能等、一部の機能により一時的に Who-Is が送信される場合があります。	3
	WhoisMasterDeviceNo	Who-Is のマスターとなっているデバイスのインスタンス番号を設定します。 0x3FFFFFFF 以上は設定なしになります。 ※IEIESupportMode が 2, 6 の時のみ有効	4294967295
	WhoisBackupDeviceNo	Who-Is のバックアップとなっているデバイスのインスタンス番号を設定します。 0x3FFFFFFF 以上は設定なしになります。 ※IEIESupportMode が 2, 6 の時のみ有効	4294967295
	WhoisRangeSendInterval	分割された Who-Is が送信される間隔を秒単位で設定します。 定期 Who-Is と、負荷軽減などのための Who-Is の分割の両方が設定されている場合に適用されます。 設定された間隔をあけて送信した結果、WhoisSendInterval の周期以内にすべてが送信し終わらない場合、未送信の Who-Is は送信されず、次の周期に最初の Who-Is から送信されますのでご注意ください。 ※IEIESupportMode が 2, 6 の時のみ有効	0
	IEIESupportMode	BACnet の規格を設定します。 0: ANSI/ASHRAE Standard 135-2004 1: IEIEJ-P-0003:2000 アデンダム a 2: IEIEJ-G-0006:2006 アデンダム a 3: IEIEJ-P-0003:2000 4: ANSI/ASHRAE Standard 135-2010 5: ANSI/ASHRAE Standard 135-2012 6: IEIEJ-G-0006:2017	0
	TimeSetDisable	受信した TimeSynchronization の処理方法を設定します。 UTCTimeSynchronization を受信した場合、受信した時刻と UTC_Offset による調整後の時刻がどちらも 1970/1/1 9:00:00 以降の場合のみ有効です。 0: TimeSynchronization 受信時に PC の時刻を自動的に変更する。(アプリケーションを管理者権限で実行する必要がある。) 1: TimeSynchronization を受信しても、そのパケットは無視し、PC の時刻を変更しない。	1
	TimeSyncMaster	定期的に TimeSynchronization サービスを送信するかどうかを設定します。 0:TimeSynchronization を送信しない。 1: 毎日定時に TimeSynchronization を送信する。	0
	TimeSyncHour	上記 TimeSyncMaster が 1 の場合に、TimeSynchronization リクエストを送信する時刻のうち、「時」を示します。	1
	TimeSyncMinute	上記 TimeSyncMaster が 1 の場合に、TimeSynchronization リクエストを送信する時刻のうち、「分」を示します。	15

セクション名	キー名	内容	デフォルト値
BACnet	AutoAddressBindingSetDis able	AddressBinding の登録方法を設定します。 0: DeviceFile.csv で記述されたデバイスだけでなく、受信した I-Am のデバイス全てを DeviceAddressBinding に登録する。 1:I-Am を受信しても、DeviceAddressBinding に自動登録しない。DeviceFile.csv で記述されたデバイスのみ DeviceAddressBindig に登録する。	0
	ScanStatusFlagsAutoAddDi sable	Scan で PresentValue を登録した際に、自動的に StatusFlags も登録するかを設定します。 0:自動的に StatusFlags を付加する 1:自動的に StatusFlags を付加しない	0
	ChangeOfReliabilityEvent TypeSupport	BACnet2012 で追加されたイベントタイプである ChangeOfReliability タイプの EventNotification をサポートとするかを設定します。 0: サポートしない 1:サポートする	0
	SystemStatusReadInterval	他の機器の System_Status を定期的に読み出す周期を 0-600 (秒) の間で指定します。 ただし 0 が指定されると読み出しは行いません。	0
	CheckAliveInterval	生存(Operational 状態であるか)を確認するために、他の機器からの I-Am サービスリクエストを監視する周期を「秒」単位で指定します。 ただし 0 が指定されると監視は行いません。	180
File	Device	デバイス管理対象リストファイルのパス名を指定します。	./DeviceFile.csv
	RestartNotification	RestartNotification 通知先ファイルのパス名を指定します。	./RestartNotifi cation.csv
	Scan	定期収集設定ファイルのパス名を指定します。	./Scan.csv
	WhoIs	定期 Who-Is のデバイス範囲指定用ファイルのパス名を指定します。	./WhoIsFile.csv

(2) デバイス管理対象リストファイル

このファイルは CSV ファイルであり、本製品が管理する BACnet デバイスを起動時に予め登録しておくことができます。CSV ファイルのカラム 1 と 2 は必須、カラム 3 以降を指定したい場合は必ず 6 カラム目まで記載する必要があります。カラム 3 以降を指定しない場合、通信相手デバイスから I-Am を受信すると IPAddress や PortNo が自動設定され、I-Am 監視は BACnet.ini ファイルの CheckAliveInterval の値に従います。I-Am の送信ができないデバイスと通信する場合は、本ファイルにてカラム 6 まで設定することにより通信させることができます。

カラム	内容	備考
1	DeviceInstanceNo	通信相手の DeviceInstance 番号を設定します。
2	MaxOutStandingPDUCount	1 カラム目で指定した Device に対して、そのデバイスからの応答に関係なく連続で送信可能なパケット数を指定します。
3	NetworkNo(※1)	通信相手デバイスのネットワーク番号を指定します。使用しない場合は 0
4	IPAddress(※2)	通信相手デバイスの IP アドレスを指定します。(例) 192.168.1.1
5	PortNo(※2)	通信相手デバイスのポート番号を指定します。47808 推奨
6	I-Am 監視禁止(※1)	通信相手デバイスの I-Am を監視するかを指定します。0:する 1:しない

※1 不正な値だった場合は、0 が指定される

※2 不正な値だった場合は、DeviceInstanceNo と MaxOutStandingPDUCount のみが登録される

(3) RestartNotification 通知先ファイル

このファイルは CSV ファイルであり、参入離脱時に COV 通知する相手を設定します。

カラム	内容	備考
1	DeviceInstanceNo または IP アドレスポート番号	IP アドレスポート番号の場合は 192.168.0.1:47808 の形式

(4) 定期収集設定ファイル

このファイルは **CSV** ファイルであり、定期収集の設定を行います。

カラム	内容	備考
1	DeviceInstanceNo	収集先の DeviceInstance 番号を設定します。
2	ObjectType	収集するオブジェクトタイプを設定します。
3	InstanceNo	収集するオブジェクトのインスタンス番号を設定します。
4	PropertyID	収集するプロパティ番号を設定します。
5	ArrayIndex	カラム 4 で指定した PropertyID が配列型のプロパティでない場合は、何も記述しないでください。
6	収集周期	収集する周期(秒)を指定します。
7	グループリスト	所属するグループ番号を：で区切って指定します。 グループ番号は 1 から始まる整数を指定してください。 例 1:20:30 この場合は 1 および 20、30 グループに属します。

(5) 定期 Who-Is のデバイス範囲指定用ファイル

このファイルは **CSV** ファイルであり、定期 **Who-Is** を送信する際のデバイス範囲を設定します。

カラム	内容	備考
1	LowRange	DeviceInstanceRangeLowLimit を指定します。 -1 の場合は指定なしになります。
2	HighRange	DeviceInstanceRangeHighLimit を指定します。 -1 の場合は指定なしになります。

3.7.3. BD_OpenDirect

概 要	BACnet 通信の開始(引数指定版) [オープンクローズ]	
機 能	引数で指定された内容で通信動作が開始できる状態にします。 Device オブジェクトの SystemStatus プロパティは NoOperational となります。	
書 式	<pre>int BD_OpenDirect(int NetworkNo, char* IPAddress, char* BroadCastAddress, int PortNo, ULONG DeviceInstanceNo, BYTE AnnexUEnable = 0);</pre>	
引 数	int NetworkNo	ネットワーク番号
	char* IPAddress	自身の IP アドレス（詳細は「3.7.1(1)IP アドレス」を参照）
	char* BroadCastAddress	同報アドレス（IPv6 の場合はマルチキャストアドレス）
	int PortNo	ポート番号（通常 47808）
	ULONG DeviceInstanceNo	自身の DeviceInstanceNo（詳細は「3.7.1(2)DeviceInstanceNo」を参照）
	BYTE AnnexUEnable	BACnet/IPv6（AnnexU）の使用有無(0:不使用, 1:使用) 値を指定しない場合 0
戻り値	0	正常にオープンできた。
	0 以外	オープンに失敗した。 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	本ドライバを使用する場合は、最初にこの関数または BD_Open、BD_OpenDirectB のいずれかを実行しなければいけません。	

3.7.4. BD_OpenDirectB

概 要	BACnet 通信の開始(引数指定版 3) [オープンクローズ]	
機 能	引数で指定された内容で通信動作が開始できる状態にします。 Device オブジェクトの SystemStatus プロパティは NoOperational となります。	
書 式	<pre>int BD_OpenDirectB(int NetworkNo, char* IPAddress, char* BroadCastAddress, int PortNo, ULONG DeviceInstanceNo char* FileName, BYTE AnnexUEnable = 0);</pre>	
引 数	int NetworkNo	ネットワーク番号
	char* IPAddress	自身の IP アドレス (詳細は「3.7.1(1)IP アドレス」を参照)
	char* BroadCastAddress	同報アドレス (IPv6 の場合はマルチキャストアドレス)
	int PortNo	ポート番号 (通常は 47808 を使用する)
	ULONG DeviceInstanceNo	自身の DeviceInstanceNo (詳細は「3.7.1(2)DeviceInstanceNo」を参照)
	char* FileName	「構成ファイル」の場所を完全パス名で指定します。
	BYTE AnnexUEnable	BACnet/IPv6 (AnnexU) の使用有無(0:不使用, 1:使用) 値を指定しない場合 0
戻り値	0	正常にオープンできた。
	0 以外	オープンに失敗した。 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	本ドライバを使用する場合は、最初にこの関数または BD_Open、BD_OpenDirect のいずれかを実行しなければいけません。	

3.7.5. BD_Close

概 要	BACnet 通信の終了 [オープンクローズ]	
機 能	すべてのリソースを解放します。	
書 式	<pre>int BD_Close();</pre>	
戻り値	0	正常終了
	0 以外	エラーが発生した。 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。

3.7.6. BD_AddMonitorDevice

概 要	モニタする Device の追加 [初期設定]	
機 能	デバイス管理対象 Device に Device を追加します。	
書 式	int BD_AddMonitorDevice(ULONG DeviceInstanceNo);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3.7.1(2)DeviceInstanceNo」を参照)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	デバイス管理対象リストファイルでも同様に追加されます。	

3.7.7. BD_AddRestartNotificationDevice

概 要	宛先デバイスの追加 [初期設定]	
機 能	参入離脱時に通告する宛先を指定します。 この Device に対して Who-Is でアドレスを決定してから参入離脱の COV を通知します。	
書 式	int BD_AddRestartNotificationDevice(ULONG DeviceInstanceNo);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3.7.1(2)DeviceInstanceNo」を参照)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	RestartNotification 通知先ファイルでも同様に追加されます。	

3.7.8. BD_AddRestartNotificationAddress

概 要	宛先リストへの追加 [初期設定]	
機 能	参入離脱時に通告する宛先リストに追加します。	
書 式	int BD_AddRestartNotificationAddress(char* IPAddress, int PortNo);	
引 数	char* IPAddress	IP アドレス (詳細は「3.7.1(1)IP アドレス」を参照)
	int PortNo	ポート番号(通常は 47808 を使用します)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	RestartNotification 通知先ファイルでも同様に追加されます。	

3. 7. 9. BD_AddScan

概 要	定期収集の登録 [初期設定]	
機 能	定期収集の登録を行います。本関数で登録を行った直後より定期収集を開始します。	
書 式	int BD_AddScan(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Interval, int GroupNo);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	PropertyID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	int Interval	収集周期 (単位 : 秒、0 の時は定期収集を行わない)
	int GroupNo	グループ番号 (1 以上の値を指定する)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。

3. 7. 10. BD_AddScanGroup

概 要	定期収集の追加 [初期設定]	
機 能	登録済の定期収集情報にさらなるグループ指定を追加します。	
書 式	int BD_AddScanGroup(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int GroupNo);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	PropertyID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	int GroupNo	グループ番号 (1 以上の値を指定する。)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。

3.7.11. BD_AddRemoteDevice

概 要	定期収集の追加[初期設定]	
機 能	AutoAddressBindingSetDisable=1 のとき、DeviceAddressBinding プロパティにエントリを追加するために使用します。	
書 式	int BD_AddRemoteDevice (ULONG DeviceInstanceNo);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3.7.1(2)DeviceInstanceNo」を参照)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	実際にエントリとして有効になるのは I-Am を受け取るなどして IP アドレスが判明した時点になります。	

3.7.12. BD_SetDeviceStatusChangeCallback

概 要	システム状態のコールバック関数指定 [コールバック設定]	
機 能	BACnetDevice の SystemStatus プロパティが変化すると呼ばれるコールバック関数を指定します。 この変化で、SystemStatus プロパティの COV 受信時に結果などを検出することができます。	
書 式	int BD_SetDeviceStatusChangeCallback (void* UserFunction);	
引 数	void* UserFunction	コールバック関数のポインタ (NULL ポインタを指定するとコールバックの設定が解除される)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	コールバック関数の書式は、次の通り。必ず __stdcall 指定をする必要があります。 void __stdcall UserFunction (ULONG DeviceInstanceNo, int NewSystemStatus, int BeforeSystemStatus); SystemStatus の値の詳細は「2.7SystemStatus プロパティ」を参照してください。 この関数は BD_Open 等を実行する前から実行可能です。	

3. 7. 13. BD_SetAccessCallback

概 要	通信サービスのコールバック関数指定 [コールバック設定]	
機 能	後述の通信サービス用の関数（プロパティアクセス機能）を呼びコールバックモードで1を指定した場合、通信処理が完了した時に呼ばれるコールバック関数を指定します。	
書 式	int BD_SetAccessCallback(void* UserFunction);	
引 数	void* UserFunction	コールバック関数のポインタ (NULL ポインタを指定するとコールバックの設定が解除される)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	コールバック関数の書式は、次の通り。必ず__stdcall 指定をする必要があります。 void __stdcall UserFunction(int Handle); この関数はBD_Open 等を実行する前から実行可能です。	

3. 7. 14. BD_SetScanCallback

概 要	定期収集のコールバック関数指定 [コールバック設定]	
機 能	定期収集設定したプロパティの読み込みが完了した時点で呼ばれるコールバック関数を指定します。 1 回のコールバックで読み込んだ複数のプロパティ値を同時に処理できます。	
書 式	int BD_SetScanCallback(void* UserFunction);	
引 数	void* UserFunction	コールバック関数のポインタ (NULL ポインタを指定するとコールバックの設定が解除される)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	コールバック関数の書式は、次の通り。必ず__stdcall 指定をする必要があります。 void __stdcall UserFunction(ULONG DeviceInstanceNo, ULONG ObjectID, int PropValueCount, tPropertyValue* PropertyValueList); この関数はBD_Open 等を実行する前から実行可能です。	

3. 7. 15. BD_SetScanPropertyCallback

概 要	定期収集のコールバック関数指定 [コールバック設定]	
機 能	定期収集設定したプロパティの読み込みが完了した時点で呼ばれるコールバック関数を指定します。 1回のコールバックで1つずつ個別にプロパティ値を処理できます。	
書 式	int BD_SetScanPropertyCallback(void* UserFunction);	
引 数	void* UserFunction	コールバック関数のポインタ (NULL ポインタを指定するとコールバックの設定が解除される)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	コールバック関数の書式は、次の通り。必ず__stdcall 指定をする必要があります。 void __stdcall UserFunction(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, tSimpleValue * PropertyValue); この関数はBD_Open 等を実行する前から実行可能です。	

3. 7. 16. BD_SetCOVCallback

概 要	COV 受信のコールバック関数指定 [コールバック設定]	
機 能	COV を受信すると呼ばれるコールバック関数を指定します。	
書 式	int BD_SetCOVCallback(void* UserFunction);	
引 数	void* UserFunction	コールバック関数のポインタ (NULL ポインタを指定するとコールバックの設定が解除される)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	コールバック関数の書式は、次の通り。必ず__stdcall 指定をする必要があります。 void __stdcall UserFunction(tCOVData* CovData); この関数はBD_Open 等を実行する前から実行可能です。	

3. 7. 17. BD_SetEventCallback

概 要	Event 受信のコールバック関数指定 [コールバック設定]	
機 能	Event を受信すると呼ばれるコールバック関数を指定します。	
書 式	int BD_SetEventCallback(void* UserFunction);	
引 数	void* UserFunction	コールバック関数のポインタ (NULL ポインタを指定するとコールバックの設定が解除される)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	コールバック関数の書式は、次の通り。必ず__stdcall 指定をする必要があります。 void __stdcall UserFunction(tEventData* EventData); この関数はBD_Open 等を実行する前から実行可能です。	

3. 7. 18. BD_PropertyChangeCallback

概 要	プロパティ変更時のコールバック関数指定 [コールバック設定]	
機 能	自身の BACnet オブジェクトプロパティの値が変更されると呼ばれるコールバック関数を指定します。	
書 式	int BD_PropertyChangeCallback(void* UserFunction);	
引 数	void* UserFunction	コールバック関数のポインタ (NULL ポインタを指定するとコールバックの設定が解除される)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	コールバック関数の書式は、次の通り。必ず__stdcall 指定をする必要があります。 void __stdcall UserFunction(ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority); ここで ObjectID, PropertyID, ArrayIndex で変更された BACnetObject とプロパティが示されます。 ArrayIndex は使用されていない場合は負の値が設定されます。 Priority は Priority 付きで書き込まれた場合 (通常 PriorityArray プロパティを有する BACnetObject の PresentValue に対する WriteProperty (Multiple) で指定される) の Priority です。使用されていない場合は 0 または負の値が設定されます。 この関数はBD_Open 等を実行する前から実行可能です。	

3. 7. 19. BD_GetDeviceStatus

概 要	システム状態の取得 [デバイス管理]	
機 能	BACnetDevice の SystemStatus を取得します。	
書 式	int BD_GetDeviceStatus (ULONG DeviceInstanceNo);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
戻り値	負の値	指定された DeviceInstanceNo は管理されていない。
	0	operational
	1	operational_read_only
	2	download_required
	3	download_in_progress
	4	non_operational
	5	backup_in_progress
	6	unknown
参 考	この関数は本ドライバが管理している Device の情報が対象です。 この関数で ReadProperty を発行するものではないことに注意が必要です。	

3. 7. 20. BD_GetDeviceAddressBinding

概 要	BACnet デバイスのアドレス情報の取得 [デバイス管理]	
機 能	指定した BACnetDevice に関するアドレス情報を取得します。 AddressBinding に本ドライバが保持しているのアドレス情報がコピーされます。	
書 式	int BD_GetDeviceAddressBinding (ULONG DeviceInstanceNo, tAddressBinding* AddressBinding);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	tAddressBinding* AddressBinding	AddressBinding 格納領域
戻り値	負の値	指定された DeviceInstanceNo は管理されていない。
	0	正常
参 考	この関数は本ドライバが管理している Device の情報が対象です。 この関数で ReadProperty を発行するものではないことに注意が必要です。	

3. 7. 21. BD_SendWhois

概 要	Who-Is の送信 [デバイス管理]	
機 能	デバイス管理対象 Device リストで指定されている宛先に Who-Is を送信します。	
書 式	int BD_SendWhois();	
戻り値	0	正常に設定できた
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	デバイス管理対象 Device リストがない場合、Who-Is は送信されません。	

3. 7. 22. BD_SendWhoisWithRange

概 要	Who-Is の送信 [デバイス管理]	
機 能	指定したデバイスの範囲に対して Who-Is を送信します。	
書 式	int BD_SendWhoisWithRange(ULONG LowLimit, ULONG HighLimit);	
引 数	ULONG LowLimit	送信する Who-Is の DeviceRange (LowLimit) を指定します。 0xFFFFFFFF を指定した場合は、LowLimit 指定なしとなります。
	ULONG HighLimit	送信する Who-Is の DeviceRange の (HighLimit) を指定します。 0xFFFFFFFF を指定した場合は、HighLimit 指定なしとなります。
戻り値	0	正常に設定できた
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	全デバイスを対象に Who-Is を送信したい場合は BD_SendWhoisWithRange (0xFFFFFFFF, 0xFFFFFFFF) としてください。	

3. 7. 23. BD_SendIam

概 要	I-Am の送信 [デバイス管理]	
機 能	自身の I-Am をブロードキャスト送信します。	
書 式	int BD_SendIam();	
戻り値	0	正常に設定できた
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	Who-Is 受信時はこの関数をコールしなくても自動的に I-Am の応答を行います。	

3. 7. 24. BD_GetMyDeviceInstanceNo

概 要	デバイス番号の取得 [デバイス管理]	
機 能	BACnetDevice のデバイス番号を取得します。	
書 式	int BD_GetMyDeviceInstanceNo (ULONG* MyDeviceInstanceNo);	
引 数	ULONG* MyDeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
戻り値	0	正常に取得できた
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 25. BD_StartStopScan

概 要	定期収集の制御 [定期収集]	
機 能	指定した定期収集グループの収集動作を制御します。	
書 式	int BD_StartStopScan(int GroupNo, int Start);	
引 数	int GroupNo	グループ番号(0 はすべてのグループを意味する)
	int Start	Start/Stop を指定(0 で停止、1 で開始)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	該当するデバイス管理設定が存在する場合も 0 を返します。	

3. 7. 26. BD_ReadScanData

概 要	データ収集の取り込み [定期収集]	
機 能	データ収集で読み込んだデータを取り込みます。	
書 式	<pre>int BD_ReadScanData (ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, tSimpleValue* ReadData);</pre>	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ObjectID	ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	int PropertyID	PropertyID (詳細は「3. 7. 1 (4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1 (5) ArrayIndex」を参照)
	tSimpleValue* ReadData	読み込みデータ
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 27. BD_ReadScanMultipleData

概 要	データ収集の取り込み(マルチプル) [定期収集]	
機 能	データ収集で読み込んだデータを取り込みます。	
書 式	<pre>int BD_ReadScanMultipleData (ULONG DeviceInstanceNo, ULONG ObjectID, int SpecCount, tPropertyValue* ReadSpec);</pre>	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	int SpecCount	PropertyValue の数
	tPropertyValue* ReadSpec	tPropertyValue 領域の先頭アドレス
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	読み出そうとしたデータが存在しなかった場合、ReadSpec には ErrorClass : Property, ErrorCode : ErrorOther がセットされます。また、個別のデータにエラーが含まれている場合でも関数自体は成功となります。	

3. 7. 28. BD_SetSystemStatus

概 要	システム情報の設定 [自デバイス制御]	
機 能	自身の Device の SystemStatus を指定された値に設定し、その値を BACnet システムへの参入動作のため COV 通告、または Event を同報します。 SystemStatus として設定された値により自身の DeviceObject の SystemStatus を変更すると共に RestartNotification 通知先に COV (SystemStatus) を送信します。	
書 式	int BD_SetSystemStatus(int SystemStatus);	
引 数	int SystemStatus	自身の Device の SystemStatus (詳細は「2. 7SystemStatus プロパティ」を参照)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	この関数を使用して SystemStatus=6(Unknown)に設定することはできません。	

3. 7. 29. BD_SetMaxOutStandingPDUCount

概 要	連続で送信可能なパケット数の設定 [自デバイス制御]	
機 能	DeviceInstanceNo で指定した Device に対して、そのデバイスからの応答に関係なく連続で送信可能なパケット数を指定します。	
書 式	int BD_SetMaxOutStandingPDUCount (ULONG DeviceInstanceNo, ULONG MaxOutStandingPduCount);	
引 数	ULONG DeviceInstanceNo	DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG MaxOutStandingPduCount	MaxOutStandingPDUCount (0~2147483647) (詳細は「3. 7. 2 (2) デバイス管理対象リストファイル」を参照)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	AddressBinding に含まれていない DeviceInstanceNo を指定した場合、BD_ERROR_NODATA (-6) を返します。 MaxOutStandingPduCount は 0 を指定すると送信数の制御を行わず、1~2147483647 までを指定すると送信数をその数までに制御します。	

3. 7. 30. BD_SetWhoIsRangeSendInterval

概 要	分割された Who-Is が送信される間隔の設定 [自デバイス制御]	
機 能	定期的な Who-Is の送信と、Who-Is の分割の両方が設定されている場合の、分割された Who-Is が送信される間隔を秒単位で設定します。	
書 式	int BD_SetWhoIsRangeSendInterval (ULONG WhoIsRangeSendInterval);	
引 数	ULONG WhoIsRangeSendInterval	Who-Is 送信間隔 (単位: 秒)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 31. BD_SendTimeSync

概 要	時刻同報 [通信]	
機 能	時刻設定 (TimeSynchronization) を同報します。	
書 式	int BD_SendTimeSync (tDateTime* DateTime);	
引 数	tDateTime* DateTime	日付時刻を指定する。
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 32. BD_SendTimeSyncUnicast

概 要	時刻送信 [通信]	
機 能	指定した DeviceInstanceNo へ時刻設定 (TimeSynchronization) を Unicast 送信します。	
書 式	int BD_SendTimeSyncUnicast (tDateTime* DateTime, ULONG DeviceInstanceNo);	
引 数	tDateTime* DateTime	日付時刻を指定する。
	ULONG DeviceInstanceNo	送信先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 33. BD_SendUTCTimeSync

概 要	時刻同報 [通信]	
機 能	時刻設定 (UTCTimeSynchronization) を同報します。	
書 式	int BD_SendUTCTimeSync (tDateTime* DateTime);	
引 数	tDateTime* DateTime	日付時刻を指定する。
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 34. BD_SetSubscribeCOV

概 要	COV 周期送信 [通信]	
機 能	SubscribeCOV を指定された周期で送信します。 本ドライバは 0 以外の LifeTime が指定されると SubscribeCOV を LifeTime の半分の周期で送信します。 また LifeTime として 0 で指定すると、定期的に送信していた SubscribeCOV の送信を終了するとともに、宛先に停止を通知します。	
書 式	int BD_SetSubscribeCOV (ULONG DeviceInstanceNo, ULONG ObjectID, ULONG ProcessID, int Confirmed, ULONG LifeTime);	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	ULONG ProcessID	ProcessIdentifier
	int Confirmed	Confirmed (0 は Unconfirmed、1 は Confirmed)
	ULONG LifeTime	ライフタイム (ここに 0 が指定されると Subscribe を停止する)
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 35. BD_WriteProperty

概 要	プロパティのライト [WriteProperty 関連]	
機 能	単純なデータ型を持つBACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。	
書 式	<pre>int BD_WriteProperty(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, tSimpleValue* WriteData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先の PropertyID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	tSimpleValue* WriteData	書き込みデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 36. BD_WritePropertyUnsigned

概 要	プロパティのライト(符号なし整数型) [WriteProperty 関連]	
機 能	符号なし整数型を持つBACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。	
書 式	<pre>int BD_WritePropertyUnsigned(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, ULONG WriteData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先の PropertyID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	ULONG WriteData	書き込みデータ(符号なし整数値)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 37. BD_WritePropertyEnumerated

概 要	プロパティのライト(列挙型) [WriteProperty 関連]	
機 能	列挙型を持つBACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。 列挙型を持つプロパティとしてはBinary オブジェクトの PresentValue などがあります。	
書 式	<pre>int BD_WritePropertyEnumerated(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, ULONG WriteData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先の PropertyID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	ULONG WriteData	書き込みデータ (列挙値)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 38. BD_WritePropertyReal

概 要	プロパティのライト(単精度浮動小数点数型) [WriteProperty 関連]	
機 能	単精度浮動小数点数型を持つ BACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。	
書 式	<pre>int BD_WritePropertyReal (ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, float WriteData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	int PropertyID	宛先の PropertyID (詳細は「3. 7. 1 (4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1 (5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	float WriteData	書き込みデータ (単精度浮動小数点数型)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3.7.39. BD_WritePropertyBool

概 要	プロパティのライト(論理型) [WriteProperty 関連]	
機 能	論理型を持つ BACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。	
書 式	<pre>int BD_WritePropertyBool (ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, int WriteData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3.7.1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3.7.1(3) ObjectID」を参照)
	int PropertyID	宛先の PropertyID (詳細は「3.7.1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3.7.1(5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	int WriteData	書き込みデータ (0:False, 1:True)
	tReturnCode* Result	終了コード (詳細は「3.7.1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3.7.1(6) コールバックモード」を参照)
戻り値	詳細は「3.7.1(6) コールバックモード」を参照。	

3. 7. 40. BD_WritePropertyCharacterString

概 要	プロパティのライト(文字列型) [WriteProperty 関連]	
機 能	文字列型を持つ BACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。	
書 式	<pre>int BD_WritePropertyCharacterString(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, int CharacterCode, int CodePage, int Length, char* StrData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1 (4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1 (5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	int CharacterCode	文字コード (1 を使用すること) 0:ANSIX3.4, 1:DBCS, 2:JISC6226, 3:ISO10646 (UCS-4), 4:ISO10646 (UCS-2), 5:ISO-8859-1
	int CodePage	コードページ (文字コードが 1 の場合にのみ有効) 932 を使用することを推奨。範囲は 0~65535
	int Length	文字列バイト長 (StrData の有効バイト数)
	char* StrData	書き込みデータ (文字列の先頭アドレスのポインタ) データの途中に NULL が含まれる場合でも Length 変数で指定された長さのデータとして扱われます。
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 41. BD_WritePropertyEventTransitionBits

概 要	プロパティのライト(EventTransitionBits 型) [WriteProperty 関連]	
機 能	EventTransitionBits 型を持つ BACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。この型は主に EventEnable などで使用されています。	
書 式	<pre>int BD_WritePropertyEventTransitionBits(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, tEventTransitionBits* BitData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1 (4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1 (5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	tEventTransitionBits* BitData	tEventTransitionBits 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 42. BD_WritePropertyLimitEnable

概 要	プロパティのライト(LimiEnable 型) [WriteProperty 関連]	
機 能	LimiEnable 型を持つ BACnet プロパティに対して WriteProperty を発行し、指定された値で書き込みを行います。	
書 式	<pre>int BD_WritePropertyLimitEnable(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, tLimiEnable* BitData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	tLimiEnable* BitData	tLimiEnable 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 43. BD_WritePropertyMultiple

概 要	プロパティのライト(マルチプル) [WriteProperty 関連]	
機 能	単純なデータ型を持つ BACnet プロパティに対して WritePropertyMultiple を発行し、指定された値で書き込みを行います。この関数では1つの Object に限り複数のプロパティの書き込みが可能です。	
書 式	<pre>int BD_WritePropertyMultiple(ULONG DeviceInstanceNo, ULONG ObjectID, int SpecCount, tWriteAccessSpec* WriteSpec, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int SpecCount	WriteSpec の数
	tWriteAccessSpec* WriteSpec	tWriteAccessSpec 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 44. BD_WritePropertyMultipleA

概 要	プロパティのライト(マルチプル) [WriteProperty 関連]	
機 能	単純なデータ型を持つ BACnet プロパティに対して WritePropertyMultiple を発行し、指定された値で書き込みを行います。この関数では1パケットで複数の Object への書き込みが可能です。	
書 式	<pre>int BD_WritePropertyMultipleA(ULONG DeviceInstanceNo, ULONG* ObjectID, int SpecCount, tWriteAccessSpec* WriteSpec, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG* ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照) の先頭アドレス。先頭アドレスから SpecCount で指定した数だけ、ObjectID を指定することができる。(先頭アドレスから指定した数×4Byte の連続領域が必要。)
	int SpecCount	ObjectID および WriteSpec の数
	tWriteAccessSpec* WriteSpec	tWriteAccessSpec 領域の先頭アドレス。この WriteSpec 数は上記 ObjectID 数と同じ数だけ領域を用意しておく必要がある。
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

サンプルプログラム

// 1パケットでDevice-1のA0-0からA0-29のPresentValueにPriority=8で値を書き込む

```
ULONG DeviceInstanceNo = 1;
ULONG ObjectID[30];
int SpecCount = 30;
tWriteAccessSpec WriteSpec[30];
tReturnCode Result;
int CallbackMode = 0;
float FloatValue = 1.2;
for (int I = 0; I < SpecCount; i++)
{
    ObjectID[i] = 1 << 22;
    ObjectID[i] += i;
    WriteSpec[i].Priority = 8;
    WriteSpec[i].DataValue.PropertyID = 85;
    WriteSpec[i].DataValue.ArrayIndex = -1;
    WriteSpec[i].DataValue.DataValue.DataType = 4;
    WriteSpec[i].DataValue.DataValue.RealData = FloatValue * (i + 1);
}
BD_WritePropertyMultipleA(DeviceInstanceNo, &ObjectID[0], SpecCount, &WriteSpec[0], &Result, CallbackMode);
```

3. 7. 45. BD_ReadProperty

概 要	プロパティのリード [ReadProperty 関連]	
機 能	単純なデータ型を持つBACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadProperty(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, tSimpleValue* ReadData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	tSimpleValue* ReadData	読み込みデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 46. BD_ReadPropertyUnsigned

概 要	プロパティのリード(符号なし整数型) [ReadProperty 関連]	
機 能	符号なし整数型を持つBACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyUnsigned(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, ULONG* ReadData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	ULONG* ReadData	読み込みデータ (符号なし整数値)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 47. BD_ReadPropertyEnumerated

概 要	プロパティのリード(列挙型) [ReadProperty 関連]	
機 能	列挙型を持つ BACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyEnumerated(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, ULONG* ReadData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	ULONG* ReadData	読み込みデータ (符号なし整数値)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 48. BD_ReadPropertyReal

概 要	プロパティのリード(単精度浮動小数点型) [ReadProperty 関連]	
機 能	単精度浮動小数点型を持つ BACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyReal(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, float* ReadData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	float* ReadData	読み込みデータ (単精度浮動小数点型)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 49. BD_ReadPropertyUnsignedWithStatus

概 要	プロパティのリード(符号なし整数型) [ReadProperty 関連]	
機 能	符号なし整数型を持つ BACnet プロパティと同じ ObjectID の StatusFlags プロパティに対して ReadPropertyMultiple を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyUnsignedWithStatus(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, ULONG* ReadData, tStatusFlags* StatusFlags, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	ULONG* ReadData	読み込みデータ (符号なし整数値)
	tStatusFlags* StatusFlags	StatusFlags 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
戻り値	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 50. BD_ReadPropertyEnumeratedWithStatus

概 要	プロパティのリード(列挙型) [ReadProperty 関連]	
機 能	列挙型を持つ BACnet プロパティと同じ ObjectID の StatusFlags プロパティに対して ReadPropertyMultiple を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyEnumeratedWithStatus(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, ULONG* ReadData, tStatusFlags* StatusFlags, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4) PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5) ArrayIndex」を参照)
	ULONG* ReadData	読み込みデータ (列挙値)
	tStatusFlags* StatusFlags	StatusFlags 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
戻り値	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 51. BD_ReadPropertyRealWithStatus

概 要	プロパティのリード(単精度浮動小数点数型) [ReadProperty 関連]	
機 能	単精度浮動小数点数型を持つ BACnet プロパティと同じ ObjectID の StatusFlags プロパティに対して ReadPropertyMultiple を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyRealWithStatus(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, float* ReadData, tStatusFlags* StatusFlags, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	float* ReadData	読み込みデータ (単精度浮動小数点数型)
	tStatusFlags* StatusFlags	StatusFlags 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6)コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6)コールバックモード」を参照。	

3. 7. 52. BD_ReadPropertyBool

概 要	プロパティのリード(論理型) [ReadProperty 関連]	
機 能	論理型を持つ BACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyBool(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int* ReadData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	int* ReadData	読み込みデータ (0:False, 1:True)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6)コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6)コールバックモード」を参照。	

3. 7. 53. BD_ReadPropertyCharacterString

概 要	プロパティのリード(文字列型) [ReadProperty 関連]	
機 能	文字列型を持つ BACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyCharacterString(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int* CharacterCode, int* CodePage, int BufferLength, int* Length, char* StrData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	int* CharacterCode	文字コード (1 を使用すること) 0:ANSIX3.4, 1:DBCS, 2:JISC6226, 3:ISO10646 (UCS-4), 4:ISO10646 (UCS-2), 5:ISO-8859-1
	int* CodePage	コードページ (文字コードが 1 の場合にのみ有効) 932 を使用することを推奨。範囲は 0~65535
	int BufferLength	読み込みバッファのバイト数
	int* Length	StrData 内に読み込まれた文字列の有効バイト数。
	char* StrData	読み込みバッファ先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6)コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6)コールバックモード」を参照。 読み込んだデータ長が BufferLength 変数で指定された値を超える場合、Length 変数は BufferLength より 1 小さい値となり、StrData の終端に NULL が挿入されます。	

3. 7. 54. BD_ReadPropertyEventTransitionBits

概 要	プロパティのリード(EventTransitionBits 型) [ReadProperty 関連]	
機 能	EventTransitionBits 型を持つ BACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyEventTransitionBits(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, tEventTransitionBits* BitData, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	tEventTransitionBits* BitData	tEventTransitionBits 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6)コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6)コールバックモード」を参照。	

3. 7. 55. BD_ReadPropertyLimitEnable

概 要	プロパティのリード(LimitEnable 型) [ReadProperty 関連]	
機 能	LimitEnable 型を持つ BACnet プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyLimitEnable(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, tLimitEnable* LimitEnable, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	tLimitEnable* LimitEnable	tLimitEnable 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6)コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6)コールバックモード」を参照。	

3. 7. 56. BD_ReadPropertyStatusFlags

概 要	プロパティのリード(システム状態) [ReadProperty 関連]	
機 能	指定された ObjectID の StatusFlags プロパティに対して ReadProperty を発行し、読み込みを行います。	
書 式	<pre>int BD_ReadPropertyStatusFlags(ULONG DeviceInstanceNo, ULONG ObjectID, tStatusFlags* StatusFlags, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3) ObjectID」を参照)
	tStatusFlags* StatusFlags	StatusFlags 型のデータ
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 57. BD_ReadPropertyMultiple

概 要	プロパティのリード(マルチプル) [ReadProperty 関連]	
機 能	単純なデータ型を持つ BACnet プロパティに対して ReadPropertyMultiple を発行し、読み込みを行います。この関数では 1 つの Object に限り複数のプロパティの読み込みが可能です。	
書 式	<pre>int BD_ReadPropertyMultiple(ULONG DeviceInstanceNo, ULONG ObjectID, int SpecCount, tPropertyValue* ReadSpec, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	int SpecCount	ReadSpec の数
	tPropertyValue* ReadSpec	tPropertyValue 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 58. BD_ReadPropertyMultipleA

概 要	プロパティのリード(マルチプル) [ReadProperty 関連]	
機 能	単純なデータ型を持つ BACnet プロパティに対して ReadPropertyMultiple を発行し、読み込みを行います。この関数では複数の Object に複数のプロパティ読み込みが可能です。	
書 式	<pre>int BD_ReadPropertyMultipleA(ULONG DeviceInstanceNo, ULONG* ObjectID, int SpecCount, tPropertyValue* ReadSpec, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG* ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照) の先頭アドレス。先頭アドレスから SpecCount で指定した数だけ、ObjectID を指定することができます。(先頭アドレスから指定した数×4Byte の連続領域が必要。)
	int SpecCount	ObjectID および ReadSpec の数
	tPropertyValue* ReadSpec	tPropertyValue 領域の先頭アドレス。この PropertyValue 数は上記 ObjectID 数と同じ数だけ領域を用意しておく必要がある。
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 59. BD_AddListElement

概 要	リスト追加 [List 操作]	
機 能	単純データ型のリスト構造をもつプロパティに対して AddListElement サービスを発行し、リスト要素を追加します。	
書 式	<pre>int BD_AddListElement(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int SpecCount, tSimpleValue* ListSpec, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	int SpecCount	ListSpec の数
	tSimpleValue* ListSpec	tSimpleValue 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6)コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6)コールバックモード」を参照。	

3. 7. 60. BD_RemoveListElement

概 要	リスト削除 [List 操作]	
機 能	単純データ型のリスト構造をもつプロパティに対して RemoveListElement サービスを発行し、リスト要素を削除します。	
書 式	<pre>int BD_RemoveListElement(ULONG DeviceInstanceNo, ULONG ObjectID, int PropertyID, int ArrayIndex, int SpecCount, tSimpleValue* ListSpec, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2)DeviceInstanceNo」を参照)
	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	int SpecCount	ListSpec の数
	tSimpleValue* ListSpec	tSimpleValue 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6)コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6)コールバックモード」を参照。	

3. 7. 61. BD_WriteScheduleObjectPropertyList

概 要	Schedule プロパティへのライト [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ListOfObjectPropertyReferences プロパティに WriteProperty で指定された情報を書き込みます。	
書 式	int BD_WriteScheduleObjectPropertyList (ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int ListCount, tObjectProperty* WriteList, tReturnCode* Result, int CallbackMode);	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int ListCount	WriteList の数
	tObjectProperty* WriteList	tObjectProperty 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 62. BD_ReadScheduleObjectPropertyList

概 要	Schedule プロパティのリード [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ListOfObjectPropertyReferences プロパティを ReadProperty で読み込みます。	
書 式	int BD_ReadScheduleObjectPropertyList (ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int BufCount, int* ReadCount, tObjectProperty* ReadBuffer, tReturnCode* Result, int CallbackMode);	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int BufCount	ReadBuffer 内に格納可能な tObjectProperty 型のデータ数
	int* ReadCount	ReadBuffer 内に読み込まれた tObjectProperty 型のデータ数
	tObjectProperty* ReadBuffer	tObjectProperty 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 63. BD_AddScheduleObjectPropertyList

概 要	Schedule プロパティの追加 [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ListOfObjectPropertyReferences プロパティに AddListElement で指定された情報を追加します。	
書 式	<pre>int BD_AddScheduleObjectPropertyList (ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int ListCount, tObjectProperty* WriteList, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFFF)
	int ListCount	WriteList の数
	tObjectProperty* WriteList	tObjectProperty 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 64. BD_RemoveScheduleObjectPropertyList

概 要	Schedule プロパティの削除 [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ListOfObjectPropertyReferences プロパティに RemoveListElement で指定された情報を削除します。	
書 式	<pre>int BD_RemoveScheduleObjectPropertyList (ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int ListCount, tObjectProperty* WriteList, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFFF)
	int ListCount	WriteList の数
	tObjectProperty* WriteList	tObjectProperty 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 65. BD_WriteWeeklySchedule

概 要	Schedule 曜日のライト [Schedule オブジェクト]	
機 能	Schedule オブジェクトの WeeklySchedule プロパティに WriteProperty で指定された情報を書き込みます。 この関数では1つの曜日のみ書き込みます。	
書 式	<pre>int BD_WriteWeeklySchedule(ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int Week, int ListCount, tTimeValue* WriteList, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int Week	曜日 1 (Monday), 2 (Tuesday) ～ 7 (Sunday)
	int ListCount	WriteList の数
	tTimeValue* WriteList	tTimeValue 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 66. BD_ReadWeeklySchedule

概 要	Schedule 曜日のリード [Schedule オブジェクト]	
機 能	Schedule オブジェクトの WeeklySchedule プロパティを ReadProperty で1つの曜日のみ読み込みます。	
書 式	<pre>int BD_ReadWeeklySchedule(ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int Week, int BufCount, int* ReadCount, tTimeValue* ReadBuffer, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int Week	曜日 1 (Monday), 2 (Tuesday) ～ 7 (Sunday)
	int BufCount	ReadBuffer 内に格納可能な tTimeValue 型のデータ数
	int* ReadCount	ReadBuffer 内に読み込まれた tTimeValue 型のデータ数
	tTimeValue* ReadBuffer	tTimeValue 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 67. BD_WriteExceptionSchedule

概 要	Schedule 配列のライト [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ExceptionSchedule プロパティに WriteProperty で指定された情報を 1 つの配列要素にのみ書き込みます。	
書 式	<pre>int BD_WriteExceptionSchedule(ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int ArrayIndex, tPeriod* PeriodData, int ListCount, tTimeValue* WriteList, int EventPriority, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1 (5) ArrayIndex」を参照) 0 および負の値を指定することはできません。
	tPeriod* PeriodData	tPeriod 領域のアドレス
	int ListCount	WriteList の数
	tTimeValue* WriteList	tTimeValue 領域の先頭アドレス
	int EventPriority	ExceptionSchedule の EventPriority (1~16)
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
戻り値	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 68. BD_ReadExceptionSchedule

概 要	Schedule 配列のリード [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ExceptionSchedule プロパティを ReadProperty で 1 つの配列要素のみ読み込む。	
書 式	<pre>int BD_ReadExceptionSchedule (ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, int ArrayIndex, tPeriod* PeriodData, int BufCount, int* ReadCount, tTimeValue* ReadBuffer, int* EventPriority, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1 (5) ArrayIndex」を参照) 0 および負の値を指定することはできません。
	tPeriod* PeriodData	tPeriod 領域のアドレス
	int BufCount	ReadBuffer 内に格納可能な tTimeValue 型のデータ数
	int* ReadCount	ReadBuffer 内に読み込まれた tTimeValue 型のデータ数
	tTimeValue* ReadBuffer	tTimeValue 領域の先頭アドレス
	int* EventPriority	ExceptionSchedule の EventPriority
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 69. BD_WriteExceptionScheduleForSevenDays

概 要	ExceptionSchedule 7 配列のライト [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ExceptionSchedule プロパティ Index 1～7 に対して WritePropertyMultiple で指定された情報を一括で書き込みます。	
書 式	<pre>int BD_WriteExceptionScheduleForSevenDays(ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, tPeriod* PeriodData, int ListCount, tTimeValue* WriteList, int* EventPriority, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	tPeriod* PeriodData	連続した 7 要素分の PeriodData 領域の先頭アドレス。Index[1]～[7]の Period を格納する。必ず sizeof(tPeriod)×7 の領域を確保しなければならない。 PeriodData[0] : Index1、・・・、PeriodData[6] : Index7
	int ListCount	WriteList で確保した領域のうち、格納可能な 1 要素分の TimeValue 数を指定する。WriteList は ListCount×7 要素分の領域を確保する必要がある。
	tTimeValue* WriteList	TimeValue リストを格納するための 7 要素分の連続した WriteList 領域の先頭アドレス。ListCount×7 要素分の領域を確保しておく必要がある。 確保された領域は以下のような割り当てとなる。 WriteList[0]～[ListCount-1] : Index1 の TimeValue 領域 WriteList[ListCount]～[ListCount*2-1] : Index2 の TimeValue 領域 ・・・ WriteList[ListCount*6]～[ListCount*7-1] : Index7 の TimeValue 領域 ※使用しない TimeValue 領域は DataValue.DataType を-1 に設定すること。
	int* EventPriority	連続した 7 要素分の EventPriority 領域の先頭アドレス。 ExceptionSchedule Index[1]～[7]の EventPriority の値を指定する (1～16)。必ず sizeof(int)×7 の領域を確保しなければならない。 EventPriority[0] : Index1、・・・、EventPriority[6] : Index7
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 70. BD_ReadExceptionScheduleForSevenDays

概 要	ExceptionSchedule 7 配列のリード [Schedule オブジェクト]	
機 能	Schedule オブジェクトの ExceptionSchedule プロパティ Index 1～7 を ReadPropertyMultiple で一括読み込みする。	
書 式	<pre>int BD_ReadExceptionScheduleForSevenDays(ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, tPeriod* PeriodData, int BufCount, tTimeValue* ReadBuffer, int* EventPriority, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	tPeriod* PeriodData	連続した 7 要素分の PeriodData 領域の先頭アドレス。Index[1]～[7]の Period の読み出し結果が格納される。必ず sizeof(tPeriod) × 7 の領域を確保しなければいけない。 読み出しができていない場合、Type に-1 がセットされる。 PeriodData[0] : Index1、・・・、PeriodData[6] : Index7
	int BufCount	ReadBuffer で確保した領域のうち、格納可能な 1 要素分の TimeValue 数を指定する。ReadBuffer は BufCount × 7 要素分の領域を確保する必要がある。
	tTimeValue* ReadBuffer	連続した 7 要素分の tTimeValue 領域の先頭アドレス。読み出した TimeValue の値が格納される。BufCount × 7 要素分の領域を確保しておく必要がある。 各 Index において読み出した TimeValue 数が BufCount より多かった場合、読み出し順で BufCount 分だけ格納され、残りは破棄される。 確保された領域は以下のような割り当てで使用される。 ReadBuffer [0]～[BufCount-1] : Index1 の TimeValue 領域 ReadBuffer [BufCount]～[BufCount*2-1] : Index2 の TimeValue 領域 ・・・ ReadBuffer [BufCount*6]～[BufCount*7-1] : Index7 の TimeValue 領域 ※読み出し結果がエラーとなった場合は DataValue.DataType に 255 または 256 が格納され、使用されなかった領域は DataValue.DataType に-1 がセットされる。
	int* EventPriority	連続した 7 要素分の EventPriority 領域の先頭アドレス。 ExceptionSchedule Index[1]～[7]の EventPriority の値が格納される。 必ず sizeof(int) × 7 の領域を確保しなければいけない。 読み出しができていない場合、-1 がセットされる。 EventPriority[0] : Index1、・・・、EventPriority[6] : Index7
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 71. BD_WriteScheduleEffectivePeriod

概 要	Schedule プロパティのライト [Schedule オブジェクト]	
機 能	Schedule オブジェクトの EffectivePeriod プロパティに WriteProperty で指定された情報を書き込みます。	
書 式	int BD_WriteScheduleEffectivePeriod(ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, tDateRange* WriteData, tReturnCode* Result, int CallbackMode);	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	tDateRange* WriteData	tDateRange 領域のアドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 72. BD_ReadScheduleEffectivePeriod

概 要	Schedule プロパティのリード [Schedule オブジェクト]	
機 能	Schedule オブジェクトの EffectivePeriod プロパティに ReadProperty で指定された情報を読み込む。	
書 式	int BD_ReadScheduleEffectivePeriod(ULONG DeviceInstanceNo, ULONG ScheduleInstanceNo, tDateRange* ReadData, tReturnCode* Result, int CallbackMode);	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG ScheduleInstanceNo	Schedule オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	tDateRange* ReadData	tDateRange 領域のアドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3.7.73. BD_WriteDateList

概 要	Calendar プロパティのライト [Calendar オブジェクト]	
機 能	Calendar オブジェクトの DateList プロパティに WriteProperty で指定された情報を書き込みます。	
書 式	<pre>int BD_WriteDateList(ULONG DeviceInstanceNo, ULONG CalendarInstanceNo, int ListCount, tCalendarEntry* WriteList, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3.7.1(2)DeviceInstanceNo」を参照)
	ULONG CalendarInstanceNo	Calendar オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int ListCount	WriteList の数
	tCalendarEntry* WriteList	tCalendarEntry 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3.7.1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3.7.1(6)コールバックモード」を参照)
戻り値	詳細は「3.7.1(6)コールバックモード」を参照。	

3.7.74. BD_ReadDateList

概 要	Calendar プロパティのリード [Calendar オブジェクト]	
機 能	Calendar オブジェクトの DateList プロパティを ReadProperty で読み込みます。	
書 式	<pre>int BD_ReadDateList(ULONG DeviceInstanceNo, ULONG CalendarInstanceNo, int BufCount, int* ReadCount, tCalendarEntry* ReadBuffer, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3.7.1(2)DeviceInstanceNo」を参照)
	ULONG CalendarInstanceNo	Calendar オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int BufCount	ReadBuffer 内に格納可能な tCalendarEntry 型のデータ数
	int* ReadCount	ReadBuffer 内に読み込まれた tCalendarEntry 型のデータ数
	tCalendarEntry* ReadBuffer	tCalendarEntry 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3.7.1(9)終了コード」を参照)
戻り値	int CallbackMode	コールバックモード (詳細は「3.7.1(6)コールバックモード」を参照)
	詳細は「3.7.1(6)コールバックモード」を参照。	

3. 7. 75. BD_AddDateList

概 要	Calendar プロパティの追加 [Calendar オブジェクト]	
機 能	Calendar オブジェクトの DateList プロパティに AddListElement で指定された情報を追加します。	
書 式	<pre>int BD_AddDateList(ULONG DeviceInstanceNo, ULONG CalendarInstanceNo, int ListCount, tCalendarEntry* AddList, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG CalendarInstanceNo	Calendar オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int ListCount	AddList の数
	tCalendarEntry* AddList	tCalendarEntry 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 76. BD_RemoveDateList

概 要	Calendar プロパティの削除 [Calendar オブジェクト]	
機 能	Calendar オブジェクトの DateList プロパティから RemoveListElement で指定された情報を削除します。	
書 式	<pre>int BD_RemoveDateList(ULONG DeviceInstanceNo, ULONG CalendarInstanceNo, int ListCount, tCalendarEntry* RemoveList, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1(2) DeviceInstanceNo」を参照)
	ULONG CalendarInstanceNo	Calendar オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	int ListCount	RemoveList の数
	tCalendarEntry* RemoveList	tCalendarEntry 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1(6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1(6) コールバックモード」を参照。	

3. 7. 77. BD_ReadRangeTrendLogByPosition

概 要	TrendLog プロパティのリード [TrendLog オブジェクト]	
機 能	TrendLog オブジェクトの LogBuffer プロパティから ReadRange (ByPosition) で指定された情報を読み込みます。	
書 式	<pre>int BD_ReadRangeTrendLogByPosition(ULONG DeviceInstanceNo, ULONG TrendLogInstanceNo, ULONG ReferenceIndex, int BufferCount, int* ReadCount, tLogRecord* RecBuffer, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG TrendLogInstanceNo	TrendLog オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	ULONG ReferenceIndex	ByPosition の ReferenceIndexNo
	int BufferCount	RecBuffer 内に格納可能な tLogRecord 型のデータ数
	int* ReadCount	RecBuffer 内に読み込まれた tLogRecord 型のデータ数
	tLogRecord* RecBuffer	tLogRecord 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 78. BD_ReadRangeTrendLogBySequenceNo

概 要	TrendLog プロパティのライト [TrendLog オブジェクト]	
機 能	TrendLog オブジェクトの LogBuffer プロパティから ReadRange (BySequenceNo) で指定された情報を読み込みます。	
書 式	<pre>int BD_ReadRangeTrendLogBySequenceNo (ULONG DeviceInstanceNo, ULONG TrendLogInstanceNo, ULONG ReferenceIndex, int BufferCount, ULONG* FirstSequenceNo, int* ReadCount, tLogRecord* RecBuffer, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG TrendLogInstanceNo	TrendLog オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	ULONG ReferenceIndex	SequenceNo の ReferenceIndex
	int BufferCount	RecBuffer 内に格納可能な tLogRecord 型のデータ数
	ULONG* FirstSequenceNo	FirstSequenceNo
	int* ReadCount	RecBuffer 内に読み込まれた tLogRecord 型のデータ数
	tLogRecord* RecBuffer	tLogRecord 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
戻り値	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 79. BD_ReadRangeTrendLogByTime

概 要	TrendLog プロパティのリード [TrendLog オブジェクト]	
機 能	TrendLog オブジェクトの LogBuffer プロパティから ReadRange (ByTime) で指定された情報を読み込みます。	
書 式	<pre>int BD_ReadRangeTrendLogByTime (ULONG DeviceInstanceNo, ULONG TrendLogInstanceNo, tDateTime* ReferenceTime, int BufferCount, ULONG* FirstSequenceNo, int* ReadCount, tLogRecord* RecBuffer, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3. 7. 1 (2) DeviceInstanceNo」を参照)
	ULONG TrendLogInstanceNo	TrendLog オブジェクトの InstanceNo (0 から 0x3FFFFFF)
	tDateTime* ReferenceTime	ByTime の ReferenceTime
	int BufferCount	RecBuffer 内に格納可能な tLogRecord 型のデータ数
	ULONG* FirstSequenceNo	FirstSequenceNo
	int* ReadCount	RecBuffer 内に読み込まれた tLogRecord 型のデータ数
	tLogRecord* RecBuffer	tLogRecord 領域の先頭アドレス
	tReturnCode* Result	終了コード (詳細は「3. 7. 1 (9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3. 7. 1 (6) コールバックモード」を参照)
戻り値	詳細は「3. 7. 1 (6) コールバックモード」を参照。	

3. 7. 80. BD_AddObjectID

概 要	BACnet オブジェクトの作成 [自身のオブジェクト作成]	
機 能	自身に BACnet オブジェクトを作成します。但し、Device は除きます。 指定された内容で BACnet オブジェクトを本ドライバ内に作成します。	
書 式	tObjectHandle BD_AddObjectID(ULONG ObjectID, tSimpleCharacterString* ObjectName);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	tSimpleCharacterString* ObjectName	ObjectName プロパティの値
戻り値	NULL (0)	エラーが発生したことを意味する。
	0 以外	ObjectHandle が戻される。
参 考	本関数を使用する前には必ず BD_Open() 関数などのオープン処理を行ってください。 ObjectName は空文字列 (0 文字の文字列) を指定することはできません。	

3. 7. 81. BD_SetProperty

概 要	自プロパティの設定 [自身のプロパティアクセス]	
機 能	単純なデータ型を持つ自身の中の BACnetObject のプロパティに指定された値を設定します。	
書 式	int BD_SetProperty(ULONG ObjectID, int PropertyID, int ArrayIndex, int Priority, tSimpleValue* WriteData);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	int Priority	書き込みプライオリティ (1~16) 0 およびマイナスの値は Priority を使用しないことを意味する。
	tSimpleValue* WriteData	書き込みデータ
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。

3. 7. 82. BD_GetProperty

概 要	自プロパティの取得 [自身のプロパティアクセス]	
機 能	単純なデータ型を持つ自身の中の BACnetObject のプロパティの読み込みを行います。	
書 式	int BD_GetProperty(ULONG ObjectID, int PropertyID, int ArrayIndex, tSimpleValue* ReadData);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	tSimpleValue* ReadData	読み込みデータ
戻り値	0	正常
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。

3. 7. 83. BD_SetPropertyUnused

概 要	自プロパティの取得 [自身のプロパティアクセス]	
機 能	指定したオブジェクトの指定したプロパティの使用許可を設定します。	
書 式	int BD_SetPropertyUnused(ULONG ObjectID, USHORT PropertyID, bool UnusedFlag);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	USHORT PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	bool UnusedFlag	true : 指定したプロパティを使用禁止にする false : 指定したプロパティを使用可能にする
戻り値	0	正常に読み込めた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。
参 考	MaxPresValue、MinPresValue プロパティ以外はデフォルトで使用可能となっています。 使用禁止にしたプロパティに対して読出しを行った場合、UnknownProperty のエラー応答を行います。	

3. 7. 84. BD_SetPVAndReliabilityUnsigned

概 要	PresentValue 設定(符号なし整数型) [自身のプロパティアクセス]	
機 能	PresentValue プロパティが符号なし整数型であるオブジェクトに対し、PresentValue プロパティと Reliability プロパティを設定します。	
書 式	int BD_SetPVAndReliabilityUnsigned(tObjectHandle ObjectHandle, ULONG PresentValue, int Reliability);	
引 数	tObjectHandle ObjectHandle	ObjectHandle (詳細は「3. 7. 1(7)ObjectHandle」を参照)
	ULONG PresentValue	PresentValue プロパティの値(符号なし整数型)
	int Reliability	Reliability プロパティの値(0~255)
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。
参 考	対象となるオブジェクトはMI、MV、AC、電力デマンド監視、電力デマンド制御、発電機負荷制御です。 OutOfService=False の時のみ設定可能です。True の時はプロパティの設定は行わずに 0 を返します。	

3. 7. 85. BD_SetPVAndReliabilityBinary

概 要	PresentValue 設定(BinaryPV 型) [自身のプロパティアクセス]	
機 能	PresentValue プロパティが BinaryPV 型である BACnet オブジェクトに対し、PresentValue プロパティと Reliability プロパティを設定します。	
書 式	int BD_SetPVAndReliabilityBinary(tObjectHandle ObjectHandle, bool PresentValue, int Reliability);	
引 数	tObjectHandle ObjectHandle	ObjectHandle (詳細は「3. 7. 1(7)ObjectHandle」を参照)
	bool PresentValue	PresentValue プロパティの値 (True:Active, False:Inactive)
	int Reliability	Reliability プロパティの値 (0~255)
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。
参 考	対象となるオブジェクトはBI、BV です。 OutOfService=False の時のみプロパティを設定可能です。True の時はプロパティの設定は行わずに 0 を返します。	

3. 7. 86. BD_SetPVAndReliabilityReal

概 要	PresentValue 設定(単精度浮動小数点数型) [自身のプロパティアクセス]	
機 能	PresentValue プロパティが単精度浮動小数点数型である BACnet オブジェクトに対し、PresentValue プロパティと Reliability プロパティを設定します。	
書 式	int BD_SetPVAndReliabilityReal(tObjectHandle ObjectHandle, float PresentValue, int Reliability);	
引 数	tObjectHandle ObjectHandle	ObjectHandle(詳細は「3. 7. 1(7)ObjectHandle」を参照)
	float PresentValue	PresentValue プロパティの値 (単精度浮動小数点数型)
	int Reliability	Reliability プロパティの値 (0~255)
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。
参 考	対象となるオブジェクトはAI, AV です。 OutOfService=False の時のみプロパティを設定可能です。True の時はプロパティの設定は行わずに 0 を返します。	

3. 7. 87. BD_SetFeedBackValueAndReliabilityUnsigned

概 要	FeedbackValue 設定(符号なし整数型) [自身のプロパティアクセス]	
機 能	FeedbackValue プロパティが符号なし整数型であるオブジェクトに対し、FeedbackValue プロパティと Reliability プロパティを設定します。	
書 式	<pre>int BD_SetFeedBackValueAndReliabilityUnsigned(tObjectHandle ObjectHandle, ULONG FeedbackValue, int Reliability);</pre>	
引 数	tObjectHandle ObjectHandle	ObjectHandle (詳細は「3. 7. 1(7)ObjectHandle」を参照)
	ULONG FeedbackValue	FeedbackValue プロパティの値 (符号なし整数型)
	int Reliability	Reliability プロパティの値 (0~255)
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。
参 考	対象となるオブジェクトはM0です。 OutOfService=False の時のみプロパティを設定可能です。True の時はプロパティの設定は行わずに 0 を返します。	

3. 7. 88. BD_SetFeedBackValueAndReliabilityBinary

概 要	FeedbackValue 設定(BinaryPV 型) [自身のプロパティアクセス]	
機 能	FeedbackValue プロパティが BinaryPV 型であるオブジェクトに対し、FeedbackValue プロパティと Reliability プロパティを設定します。	
書 式	<pre>int BD_SetFeedBackValueAndReliabilityBinary(tObjectHandle ObjectHandle, bool FeedbackValue, int Reliability);</pre>	
引 数	tObjectHandle ObjectHandle	ObjectHandle (詳細は「3. 7. 1(7)ObjectHandle」を参照)
	bool FeedbackValue	FeedbackValue プロパティの値 (True:Active, False:Inactive)
	int Reliability	Reliability プロパティの値 (0~255)
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。
参 考	対象となるオブジェクトはB0です。 OutOfService=False の時のみプロパティを設定可能です。True の時はプロパティの設定は行わずに 0 を返します。	

3. 7. 89. BD_SetReliability

概 要	Reliability 設定 [自身のプロパティアクセス]	
機 能	Reliability プロパティを設定する。	
書 式	int BD_SetReliability(tObjectHandle ObjectHandle, int Reliability);	
引 数	tObjectHandle ObjectHandle	ObjectHandle (詳細は「3. 7. 1(7)ObjectHandle」を参照)
	int Reliability	Reliability プロパティの値 (0~255)
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8) リターンコード一覧」を参照。

3. 7. 90. BD_AddPulse

概 要	パルス入力値の追加 [自身のプロパティアクセス]	
機 能	Accumulator オブジェクトに対し、パルス入力を行います。パルス数が PresentValue プロパティに加算されます。	
書 式	int BD_AddPulse(tObjectHandle ObjectHandle, ULONG PulseCount);	
引 数	tObjectHandle ObjectHandle	ObjectHandle (詳細は「3. 7. 1(7)ObjectHandle」を参照)
	ULONG PulseCount	パルス数 (0~65535)
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8) リターンコード一覧」を参照。

3. 7. 91. BD_SetPropertyCharacterString

概 要	自プロパティの設定 [自身のプロパティアクセス]	
機 能	FirmwareRevision プロパティを除く文字列型プロパティに対して、指定した文字列を設定します。	
書 式	int BD_SetPropertyCharacterString(ULONG ObjectID, int PropertyID, int ArrayIndex, tSimpleCharacterString* WriteData);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	tSimpleCharacterString* WriteData	書き込みデータ
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8) リターンコード一覧」を参照。

3. 7. 92. BD_GetPropertyCharacterString

概 要	自プロパティの取得 [自身のプロパティアクセス]	
機 能	自デバイスの指定した文字列型プロパティの値を取得します。	
書 式	<pre>int BD_GetPropertyCharacterString(ULONG ObjectID, int PropertyID, int ArrayIndex, tSimpleCharacterString* ReadData);</pre>	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	int PropertyID	宛先のプロパティ ID (詳細は「3. 7. 1(4)PropertyID」を参照)
	int ArrayIndex	宛先の ArrayIndex (詳細は「3. 7. 1(5)ArrayIndex」を参照)
	tSimpleCharacterString* ReadData	読み込みデータ
戻り値	0	正常に取得できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。

3. 7. 93. BD_SetPropertyAlarmValues

概 要	自プロパティの設定 [自身のプロパティアクセス]	
機 能	自デバイスの指定したオブジェクトの AlarmValues プロパティに対して値を上書き設定します。	
書 式	<pre>int BD_SetPropertyAlarmValues(ULONG ObjectID, ULONG* WriteData, ULONG DataCount);</pre>	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1(3)ObjectID」を参照)
	ULONG* WriteData	設定する AlarmValues プロパティの値
	ULONG DataCount	設定する AlarmValues 値の数
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1(8)リターンコード一覧」を参照。
参 考	WriteData は DataCount 分の領域を予め確保しておく必要があります。	

3. 7. 94. BD_GetPropertyAlarmValues

概 要	自プロパティの取得 [自身のプロパティアクセス]	
機 能	自デバイスの指定したオブジェクトの AlarmValue プロパティの値を取得します。	
書 式	int BD_GetPropertyAlarmValues(ULONG ObjectID, ULONG MaxCount, ULONG* ReadData, ULONG* DataCount);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	ULONG MaxCount	ReadData 内に格納可能な AlarmValues プロパティ値の数
	ULONG* ReadData	読み込みデータ
	ULONG* DataCount	ReadData 内に読み込まれた AlarmValues プロパティ値の数
戻り値	0	正常に取得できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	この関数をコールする際は、ReadData に MaxCount で指定した分の領域を予め確保しておく必要があります。	

3. 7. 95. BD_SetPropertyFaultValues

概 要	自プロパティの設定 [自身のプロパティアクセス]	
機 能	自デバイスの指定したオブジェクトの FaultValues プロパティに対して値を上書き設定します。	
書 式	int BD_SetPropertyFaultValues(ULONG ObjectID, ULONG* WriteData, ULONG DataCount);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	ULONG* WriteData	設定する FaultValues プロパティの値
	ULONG DataCount	設定する FaultValues 値の数
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	WriteData は DataCount 分の領域を予め確保しておく必要があります。	

3. 7. 96. BD_GetPropertyFaultValues

概 要	自プロパティの取得 [自身のプロパティアクセス]	
機 能	自デバイスの指定したオブジェクトの FaultValue プロパティの値を取得します。	
書 式	int BD_GetPropertyFaultValues(ULONG ObjectID, ULONG MaxCount, ULONG* ReadData, ULONG* DataCount);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	ULONG MaxCount	ReadData 内に格納可能な FaultValues プロパティ値の数
	ULONG* ReadData	読み込みデータ
	ULONG* DataCount	ReadData 内に読み込まれた FaultValues プロパティ値の数
戻り値	0	正常に取得できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	この関数をコールする際は、ReadData に MaxCount で指定した分の領域を予め確保しておく必要があります。	

3. 7. 97. BD_SetPropertyRecipientList

概 要	自プロパティの設定 [自身のプロパティアクセス]	
機 能	自デバイスの指定したオブジェクトの RecipientList プロパティに対して値を上書き設定します。	
書 式	int BD_SetPropertyRecipientList(ULONG ObjectID, tRecipientList* WriteData, ULONG DataCount);	
引 数	ULONG ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	tRecipientList* WriteData	設定する RecipientList プロパティの値
	ULONG DataCount	設定する RecipientList の数
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。
参 考	WriteData は DataCount 分の領域を予め確保しておく必要があります。	

3. 7. 98. BD_SetPropertyLogDeviceObjectPropertyReference

概 要	自プロパティの設定 [自身のプロパティアクセス]	
機 能	自デバイスの指定したオブジェクトの LogDeviceObjectProperty プロパティに対して値を上書き設定します。	
書 式	int BD_SetPropertyLogDeviceObjectPropertyReference (ULONG ObjectID, tLogDeviceObjectProperty* LogDeviceObjectProperty);	
引 数	ObjectID	宛先の ObjectID (詳細は「3. 7. 1 (3) ObjectID」を参照)
	tLogDeviceObjectProperty* LogDeviceObjectProperty	設定する LogDeviceObjectProperty プロパティの値
戻り値	0	正常に設定できた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 99. BD_SaveBACnetObject

概 要	自プロパティ情報の保存 (カレンダー/スケジュールのみ) [その他]	
機 能	自デバイスが持っているカレンダーオブジェクト、スケジュールオブジェクトの情報を指定したファイル名称で保存します。	
書 式	int BD_SaveBACnetObject (char* FilePath);	
引 数	char* FilePath	バックアップファイル名称を絶対パスで指定します。
戻り値	0	正常に保存ができた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3. 7. 100. BD_SaveBACnetObjectAll

概 要	自プロパティ情報の保存[その他]	
機 能	自デバイスが持っている全オブジェクトの情報を指定したファイル名称で保存します。	
書 式	int BD_SaveBACnetObjectAll (char* FilePath);	
引 数	char* FilePath	バックアップファイル名称を絶対パスで指定します。
戻り値	0	正常に保存ができた。
	0 以外	エラー 戻り値の詳細は「3. 7. 1 (8) リターンコード一覧」を参照。

3.7.101. BD_SaveObjectCSVFile

概 要	自デバイスのオブジェクト情報リストの保存[その他]	
機 能	自デバイスが持っているオブジェクト登録情報を電気設備学会フォーマットの CSV ファイル形式で保存します。	
書 式	int BD_SaveObjectCSVFile(char* FilePath);	
引 数	char* FilePath	CSV ファイル名称を絶対パスで指定します。
戻り値	0	正常に保存ができた。
	0 以外	エラー 戻り値の詳細は「3.7.1(8) リターンコード一覧」を参照。

3.7.102. BD_LoadBACnetObject

概 要	自プロパティ情報の復元[その他]	
機 能	起動時にこの関数をコールすることにより、オブジェクト情報をバックアップした時の状態に復元することができます。	
書 式	int BD_LoadBACnetObject(char* FilePath);	
引 数	char* FilePath	バックアップファイル名称を絶対パスで指定します。 BD_SaveBACnetObject, BD_SaveBACnetObjectAll 関数で保存したファイルが対象です。
戻り値	0	正常に復元ができた。
	0 以外	エラー 戻り値の詳細は「3.7.1(8) リターンコード一覧」を参照。

3.7.103. BD_ReinitializeDevice

概 要	ReinitializeDevice 送信 [その他]	
機 能	ReinitializeDevice を送信します。	
書 式	int BD_ReinitializeDevice(ULONG DeviceInstanceNo, int ReinitializedStateOfDevice, tSimpleCharacterString* Password, tReturnCode* Result, int CallbackMode);	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3.7.1(2) DeviceInstanceNo」を参照)
	int ReinitializedStateOfDevice	StateOfDevice の値 0:coldstart, 1:warmstart, 2:startbackup, 3:endbackup 4:startrestore, 5:endrestore, 6:abortrestore
	tSimpleCharacterString* Password	NULL ポインタが渡されるとパスワードは使用しない。
	tReturnCode* Result	終了コード (詳細は「3.7.1(9) 終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3.7.1(6) コールバックモード」を参照)
戻り値	詳細は「3.7.1(6) コールバックモード」を参照。	

3.7.104. BD_DeviceCommunicationControl

概 要	DeviceCommunicationControl の送信 [その他]	
機 能	DeviceCommunicationControl を送信します。	
書 式	<pre>int BD_DeviceCommunicationControl (ULONG DeviceInstanceNo, int TimeDuration, int EnableDisable, tSimpleCharacterString* Password, tReturnCode* Result, int CallbackMode);</pre>	
引 数	ULONG DeviceInstanceNo	宛先 DeviceInstanceNo (詳細は「3.7.1(2)DeviceInstanceNo」を参照)
	int TimeDuration	負の値の場合 TimeDuration は使用されない。
	int EnableDisable	0:Enable, 1:Disable
	tSimpleCharacterString* Password	NULL ポインタが渡されるとパスワードは使用しない。
	tReturnCode* Result	終了コード (詳細は「3.7.1(9)終了コード」を参照)
	int CallbackMode	コールバックモード (詳細は「3.7.1(6)コールバックモード」を参照)
戻り値	詳細は「3.7.1(6)コールバックモード」を参照。	

3.7.105. BD_ReleaseHandle

概 要	ハンドルの解放 [その他]	
機 能	ハンドルを解放します。	
書 式	int BD_ReleaseHandle(int Handle);	
引 数	Handle	Handle はコールバックモード=1 を指定した場合に返される正の値。
戻り値	0	正常に解放できた。
	1 以上	指定されたハンドルが通信中であり、まだ処理が完了していないことを示す。
	-1 以下	エラー 戻り値の詳細は「3.7.1(8)リターンコード一覧」を参照。
参 考	<p>この関数によりハンドルを開放すると、そのハンドルに該当するプロパティアクセス機能の結果が通知されなくなります。すなわち tReturnCode 構造体の更新、tSimpleValue 構造体の更新、コールバック関数の呼び出しなどが行われなくなります。</p> <p>この関数はBD_Open 等を実行する前から実行可能です。</p>	

3.8. データ構造

3.8.1. tReturnCode

```
typedef struct
{
    int      Result;           // 0:正常, 1:Error 受信, 2:Reject 受信, 3:Abort 受信,
                                // 4:Retry オーバ, 5:宛先デバイスがNonOperational,
                                // 8: 非対応のプロパティを含む ComplexAck を受信等, 13:宛先デバイスが不明

    int      ErrorClass;       // Result が1:Error 受信のとき有効
    int      ErrorCode;        // Result が1:Error 受信のとき有効
    int      FirstFailedElementNo; // Result が1:Error 受信のとき有効
    int      RejectReason;     // Result が2:Reject 受信のとき有効
    int      AbortReason;      // Result が3:Abort 受信のとき有効
} tReturnCode;
```

3.8.2. tSimpleBitString

```
typedef struct
{
    int      BitCount;         // 最大 32Bit まで指定可能
    ULONG    BitData;          // Bit0 が BACnetBitString の Bit0 に対応
                                // 例: EventEnable の場合
                                // Bit0:ToOffNormal, Bit1:ToFault, Bit2:ToNormal となる。
} tSimpleBitString;
```

3.8.3. tSimpleCharacterString

```
typedef struct
{
    int      CharacterCode;    // 1 を使用すること
    int      CodePage;         // 932 を使用することを推奨。範囲は 0~65535
    int      Length;           // 0~511 まで
    char     StrData[512];
} tSimpleCharacterString;
```

3.8.4. tSimpleValue

```
typedef struct
{
    int            DataType;          // 0:NULL, 1:BOOLEAN, 2:Unsigned, 3:Signed, 4:Real,
                                      // 8:BitString, 9:Enumerated, 10:Date, 11:Time,
                                      // 12:ObjectID, 26:DateTime, 111:WHData, 118:Prescale
                                      // 255:Error 応答
                                      // 256:その他のエラー
    ULONG          UnsignedData;      // DataType が BOOLEAN, Unsigned, Enumerated, Error, その他の Error の場合有効
                                      // Error の場合: 上位 16bit が ErrorClass、 下位 16bit が ErrorCode
                                      // その他のエラーの場合: tReturnCode の Result と同じ値が入る
    long           SignedData;        // DataType が Signed の場合有効
    float          RealData;          // DataType が Real の場合有効
    tSimpleBitString BitString;       // DataType が BitString の場合有効
    tDate          DateData;          // DataType が Date の場合有効
    tTime          TimeData;          // DataType が Time の場合有効
    tDateTime      DateTime;          // DataType が DateTime の場合有効
    tWHData        WHData;            // DataType が WHData の場合有効
    tPrescale      PrescaleData;      // DataType が Prescale の場合有効
} tSimpleValue;
```

3.8.5. tLimitEnable

```
typedef struct
{
    int            LowLimitEnable;     // 0=False, 1=True とし、それ以外はエラーとして扱う
    int            HighLimitEnable;    // 同上
} tLimitEnable;
```

3.8.6. tEventTransitionBits

```
typedef struct
{
    int            ToOffnrml;          // 0=False, 1=True とし、それ以外はエラーとして扱う
                                      // 以下のメンバも同様
    int            ToFault;
    int            ToNormal;
} tEventTransitionBits;
```

3.8.7. tStatusFlags

```
typedef struct
{
    int            InAlarm;            // 0=False, 1=True とし、それ以外はエラーとして扱う
                                      // 以下のメンバも同様
    int            Fault;
    int            Overridden;
    int            OutofService;
} tStatusFlags;
```

3.8.8. tPropertyValue

```
typedef struct
{
    int          PropertyID;
    int          ArrayIndex;
    tSimpleValue DataValue;
} tPropertyValue; // 負の値は ArrayIndex を使用しないことを意味する。
```

3.8.9. tWriteAccessSpec

```
typedef struct
{
    tPropertyValue DataValue;
    int            Priority;
} tWriteAccessSpec; // 範囲(1~16)。0 及び負数は不使用を意味する。
```

3.8.10. tObjectProperty

```
typedef struct
{
    ULONG DeviceInstanceNo; // 0x3FFFFE より大きな値は Device を指定しないことを意味する
    ULONG ObjectID;
    int   PropertyID;
    int   ArrayIndex;
} tObjectProperty; // 負の値は ArrayIndex を使用しないことを意味する。
```

3.8.11. tTime

```
typedef struct
{
    int Hour; // 時を指定する。負の値は WildCard
    int Minute; // 分を指定する。負の値は WildCard
    int Second; // 秒を指定する。負の値は WildCard
    int Hundredths; // 1/100 秒を指定する。負の値は WildCard
} tTime;
```

3.8.12. tTimeValue

```
typedef struct
{
    tTime      TimeData;
    tSimpleValue DataValue;
} tTimeValue;
```

3.8.13. tDate

```
typedef struct
{
    int      Year;           // 年を指定する。負の値はWildCard、1900～2154 まで指定可能
    int      Month;         // 月を指定する。負の値はWildCard、1～12 まで指定可能
    int      Day;           // 日を指定する。負の値はWildCard、1～31 まで指定可能
    int      Week;          // 曜日を指定する。負の値はWildCard、
                             // 1 (Monday), 2 (Tuesday) ～7 (Sunday) まで指定可能
} tDate;
```

3.8.14. tDateTime

```
typedef struct
{
    tDate     DateData;
    tTime     TimeData;
} tDateTime;
```

3.8.15. tDateRange

```
typedef struct
{
    tDate     StartDay;      // 開始日付
    tDate     EndDay;        // 終了日付
} tDateRange;
```

3.8.16. tWeekNDay

```
typedef struct
{
    int      Month;          // 月を指定する。負の値はWildCard、1～12 まで指定可能
    int      WeekOfMonth;    // 週を指定する。負の値はWildCard
                             // 1 は1 日～7 日, 2 は8 日～14 日, 3 は15 日～21 日,
                             // 4 は22 日～28 日, 5 は29 日～31 日, 6 は月の最後の7 日
    int      DayOfWeek;      // 曜日を指定する。負の値はWildCard、
                             // 1 (Monday), 2 (Tuesday) ～7 (Sunday) まで指定可能
} tWeekNDay;
```


3.8.17. tPeriod

```
typedef struct
{
    int            Type;                // 0:CalendarObject, 1:Date, 2:DateRange, 3:WeekNDay
    ULONG          CalendarInstanceNo;  // Type=0 の場合有効
    tDate          DateValue;           // Type=1 の場合有効
    tDateRange     DateRangeValue;      // Type=2 の場合有効
    tWeekNDay      WeekNDayValue;       // Type=3 の場合有効
} tPeriod;
```

3.8.18. tCalendarEntry

```
typedef struct
{
    int            Type;                // 1:Date, 2:DateRange, 3:WeekNDay
    tDate          DateValue;           // Type=1 の場合有効
    tDateRange     DateRangeValue;      // Type=2 の場合有効
    tWeekNDay      WeekNDayValue;       // Type=3 の場合有効
} tCalendarEntry;
```

3.8.19. tCOVData

```
typedef struct
{
    ULONG          ProcessID;           // プロセス ID
    ULONG          DeviceInstanceNo;    // BACnetDevice の InstanceNo
    ULONG          ObjectID;           // BACnetObject のオブジェクト ID
    ULONG          TimeRemaining;       // 残り時間
    int            PropertyValueCount;  // PropertyValue の数
    tPropertyValue PropertyValue[10];   // PropertyValueCount 数分存在する。
} tCOVData;
```

3.8.20. tTimeStamp

```
typedef struct
{
    int            Type;                // 0:Time, 1:Unsigned, 2:DateTime
    tTime          TimeData;           // Type=0 の場合有効
    ULONG          SequenceNo;         // Type=1 の場合のみ有効 o
    tDateTime      DateTimeData;       // Type=2 の場合有効
} tTimeStamp;
```

3.8.21. tNotificationChangeOfState

```
typedef struct
{
    ULONG                NewStateType;        // PropertyState
                                                // 0:BooleanValue, 1:BinaryValue, 2:EventType, 3:Polarity,
                                                // 4:ProgramChange, 5:ProgramState, 6:ReasonForFault, 7:Reliability,
                                                // 8:State, 9:SystemStatus, 10:Units, 11:UnsignedValue,
                                                // 12:LifeSafetyMode, 13:LifeSafetyState

    ULONG                StateData;
    tStatusFlags         StatusFlags;
} tNotificationChangeOfState;
```

3.8.22. tNotificationChangeOfValue

```
typedef struct
{
    ULONG                Type;                // 0:BitString, 1:単精度浮動小数点数型
    tSimpleBitString     BitString;
    float                RealValue;
    tStatusFlags         StatusFlags;
} tNotificationChangeOfValue;
```

3.8.23. tNotificationCommandFailure

```
typedef struct
{
    tSimpleValue         CommandValue;
    tStatusFlags         StatusFlags;
    tSimpleValue         FeedbackValue;
} tNotificationCommandFailure;
```

3.8.24. tNotificationFloatingLimit

```
typedef struct
{
    float                ReferenceValue;
    float                SetpointValue;
    float                ErrorLimit;
    tStatusFlags         StatusFlags;
} tNotificationFloatingLimit;
```

3.8.25. tNotificationOutOfRange

```
typedef struct
{
    float          ExceedingValue;
    float          DeadBand;
    float          ExceededLimit;
    tStatusFlags   StatusFlags;
} tNotificationOutOfRange;
```

3.8.26. tNotificationBufferReady

```
typedef struct
{
    tObjectProperty BufferProperty;
    ULONG           PreviousNotification;
    ULONG           CurrentNotification;
} tNotificationBufferReady;
```

3.8.27. tNotificationUnsignedRange

```
typedef struct
{
    ULONG          ExceedingValue;
    ULONG          ExceededLimit;
    tStatusFlags   StatusFlags;
} tNotificationUnsignedRange;
```

3.8.28. tEventData

```
typedef struct
{
    ULONG          ProcessID;           // プロセス ID
    ULONG          DeviceInstanceNo;    // BACnetDevice の InstanceNo
    ULONG          ObjectID;           // BACnetObject のオブジェクト ID
    tTimeStamp     TimeStampData;      // タイムスタンプ
    ULONG          NotificationClass;
    ULONG          Priority;
    int            EventType;
    tSimpleCharacterString MessageText; // MessageText が存在しない場合は
                                        // Length が 0 になる。

    int            NotifyType;
    int            AckRequired;        // 負の値は AckRequired を
                                        // 受け取っていないことを意味する。

    int            FromState;          // 負の値は FromState を
                                        // 受け取っていないことを意味する。

    int            ToState;
    int            NotificationParameterType; // 負の値は NotificationParameter を
                                        // 受け取っていないことを意味する。

    tNotificationChangeOfState NotificationChangeOfState; // NotificationParameterType=1 の場合有効
    tNotificationChangeOfValue NotificationChangeOfValue; // NotificationParameterType=2 の場合有効
    tNotificationCommandFailure NotificationCommandFailure; // NotificationParameterType=3 の場合有効
    tNotificationFloatingLimit NotificationFloatingLimit; // NotificationParameterType=4 の場合有効
    tNotificationOutOfRange NotificationOutOfRange; // NotificationParameterType=5 の場合有効
    tNotificationBufferReady NotificationBufferReady; // NotificationParameterType=10 の場合有効
    tNotificationUnsignedRange NotificationUnsignedRange; // NotificationParameterType=11 の場合有効
    tNotificationChangeOfReliability NotificationChangeOfReliability; // NotificationParameterType=19 の場合有効
} tEventData;
```

3.8.29. tLogRecord

```
typedef struct
{
    tDateTime      TimeStamp;
    int            StatusFlagsEnable; // 0 の場合は StatusFlags が有効ではないことを意味する。
    tStatusFlags   StatusFlags;
    tSimpleValue   LogData;
} tLogRecord;
```

3.8.30. tAddressBinding

```
typedef struct
{
    ULONG          DeviceInstanceNo; // BACnetDevice の InstanceNo
    int            NetworkNo;
    int            MacAddressLength; // 0 の場合は MacAddress が不明であることを意味する。
    BYTE           MacAddress[18];  // IPv4 の場合は先頭 4Byte が IPAddrss、後ろ 2Byte が Port 番号、
                                    // IPv6 の場合は先頭 16Byte が IPAddrss、後ろ 2Byte が Port 番号、
                                    // これらが NetworkByteOrder で格納される。
} tAddressBinding;
```

3.8.31. tAlarmSummary

```
typedef struct
{
    ULONG          ObjectID;
    int            EventState;
    tEventTransitionBits AckedTransition;
} tAlarmSummary;
```

3.8.32. tEnrollmentSummary

```
typedef struct
{
    ULONG          ObjectID;
    int            EventType;
    int            EventState;
    int            Priority;
    int            NotificationClass;    // 負の値はNotificationClassを受け取っていないことを意味する。
} tEnrollmentSummary;
```

3.8.33. tWHData

```
typedef struct
{
    int            UlongEnable;        // IEIEJ-P-0003:2000 及び IEIEJ-P-0003:2000 アデンダム a のとき true
    float          Consumed_WH_Float;  // UlongEnable が 0 の時有効
    ULONG          Consumed_WH_ULONG;  // UlongEnable が 0 以外有効
    int            Data_State;         // 0:正常, 1:異常, 2:無効
} tWHData;
```

3.8.34. tScale

```
typedef struct
{
    int            ScaleChoice;        // 0:Real, 1:Integer
    float          Real;               // ScaleChoice=0 のとき有効
    long           Integer;            // ScaleChoice=1 のとき有効
} tScale;
```

3.8.35. tPrescale

```
typedef struct
{
    ULONG          Multiplier;
    ULONG          ModuloDivide;
} tPrescale;
```

3.8.36. tRecipient

```
typedef struct
{
    int            DeviceObjectIDUseFlag;    // 0=False, 1=True とし、それ以外はエラーとして扱う
    ULONG          DeviceObjectID;           // Device のインスタンス番号のみを指定
    int            NetworkNo;
    int            MacAddressLength;          // 0 の場合は MacAddress が不明であることを意味する
    BYTE           MacAddress[18];           // IPV4 の場合は先頭 4Byte が IPAddrss、後ろ 2Byte が Port 番号、
                                           // IPV6 の場合は先頭 16Byte が IPAddrss、後ろ 2Byte が Port 番号、
                                           // これらが NetworkByteOrder で格納される。
} tRecipient;
```

3.8.37. tDayOfWeek

```
typedef struct
{
    int            Sunday;                   // 0=False, 1=True とし、それ以外はエラーとして扱う
                                           // 以下のメンバも同様
    int            Saturday;
    int            Friday;
    int            Thursday;
    int            Wednesday;
    int            Tuesday;
    int            Monday;
} tDayOfWeek;
```

3.8.38. tRecipientList

```
typedef struct
{
    tRecipient      Recipient;
    tDayOfWeek      ValidDays;
    tTime           FromTime;
    tTime           ToTime;
    ULONG           ProcessID;
    int             IssueConfirmed;          // 0=False, 1=True とし、それ以外はエラーとして扱う
    tEventTransitionBits Transitions;
} tRecipientList;
```

3.8.39. tLogDeviceObjectProperty

```
typedef struct
{
    bool            DeviceIDUseFlag;
    ULONG           DeviceID;                // Device のインスタンス番号のみを指定
    ULONG           ObjectID;
    ULONG           PropertyID;
    int             ArrayIndex;              // 負の値は ArrayIndex を使用しないことを意味する。
} tLogDeviceObjectProperty;
```

3.8.40. tNotificationChangeOfReliability

```
typedef struct
{
    int            Reliability;
    tStatusFlags   StatusFlags;
    float          RealPresentValue;          // オブジェクトタイプ AI, AO, AV のとき有効
    ULONG          UnsignedPresentValue;      // オブジェクトタイプ BI, BO, BV, MI, MO, MV, AC のとき有効
    ULONG          FeedbackValue;             // オブジェクトタイプ BO, MO のとき有効
    ULONG          PulseRate;                 // オブジェクトタイプ AC のとき有効
} tNotificationChangeOfReliability;
```

第4章 プログラミング

4. 1. 関数の使用例

4. 1. 1. 参入離脱の例

```
01 int userFauntion()
02 {
03     BD_Open("c:/bacnet.ini")
04     BD_SetSystemStatus(2);           // DownloadRequired の COV を送信する。
05     BD_SendWhois();                 // 管理対象 Device リストファイルで指定された Device に Who-Is を送信する。
06     BD_SetSystemStatus(3);           // DownloadInProgress の COV を送信する。
07     //BD_WriteDateList();           // 必要ならばカレンダー情報を送信する。
08     BD_SetSystemStatus(0);           // Operational の COV を送信する。
09
10     /* ここから通信処理を行う。 */
11
12     /* 終了時は以下のようにする。 */
13     BD_SetSystemStatus(4);           // NonOperational の COV を送信する。
14     return 0;
15 }
```


4.2. 他のデバイスのカレンダーオブジェクトの DateList の読み込み

(1) コールバックを使用する場合

Device10 の CAL-20 の DateList をリードする例

```
01 tCalendarEntry    CalendarEntry[100];          // DateList は最大 100 個の要素が格納されているものとする。
02 int               ReadCount;
03 tReturnCode       Result;
04
05 static int __stdcall AccessCallback(int handle)
06 {
07     if (/* handle が完了待ちに登録されている時 */)
08     {
09         /* handle をキーに Read 完了待ちから情報を取り出す。
10          * 上記の場合 Result を確認し正常であれば ReadCount に読み込まれた個数、
11          * CalendarEntry に読み込みデータが格納されているので、それを使用する。
12          */
13     }
14     return 0;
15 }
16
17 int userFauntion()
18 {
19     BD_SetAccessCallback((void*)AccessCallback);
20     int Handle = BD_ReadDateList(10, 20, 100, &ReadCount, CalendarEntry, &Result, 1);
21     if (Handle <= 0)
22     {
23         /* エラー処理 */
24     }
25     else
26     {
27         /* Read 完了を待つため Handle を Read 完了待ちに登録する。 */
28     }
29     return 0;
30 }
```

(2) コールバックを使用しない場合

完了のコールバックは使用されず、終了をポーリング処理で確認します。

Device10 の CAL-20 の DateList をリードする例。

```
01 int userFauntion()
02 {
03     tCalendarEntry    CalendarEntry[100];        // DateList は最大 100 個の要素が格納されているものとする。
04     int                ReadCount;
05     tReturnCode        Result;
06
07     Result.Result=-1;
08     if (BD_ReadDateList(10, 20, 100, &ReadCount, CalendarEntry, &Result, 0) <= 0)
09     {
10         /* エラー処理 */
11     }
12     else
13     {
14         for (;;)
15         {
16             if (Result.Result == 0)
17             {
18                 /* ReadCount および CalendarEntry に読み込まれたデータの処理。 */
19                 return 0;
20             }
21             else if (Result.Result > 0)
22             {
23                 /* エラー処理。 */
24                 return 0;
25             }
26             else
27             {
28                 /* 別の処理を行う。 */
29             }
30         }
31     }
32 }
```

(3) コールバックを使用せず完了までブロックされる場合

Device10 の CAL-20 の DateList をリードする例。

```
01 int userFauntion()
02 {
03     tCalendarEntry    CalendarEntry[100];        // DateList は最大 100 個の要素が格納されているものとする。
04     int                ReadCount;
05     tReturnCode        Result;
06
07     Result.Result=-1;
08     if (BD_ReadDateList(10, 20, 100, &ReadCount, CalendarEntry, &Result, 2) != 0)
09     {
10         /* エラー処理。 */
11     }
12     else
13     {
14         if (Result.Result == 0)
15         {
16             /* ReadCount および CalendarEntry に読み込まれたデータの処理。 */
17         }
18         else
19         {
20             /* エラー処理。 */
21         }
22     }
23     return 0;
24 }
```

4. 3. 自身のオブジェクト作成

```
01  tObjectHandle    ObjectHandle;
02
03  int userFaution()
04  {
05      ULONG ObjectID;
06      tSimpleCharacterStringObjectName;
07
08      ObjectID = 3 << 22;                // BI を指定し 22Bit 目まで左シフトする
09      ObjectID |= 123;                  // BI-123
10      ObjectName.CharacterCode = 1;    // DBCS
11      ObjectName.CodePage = 932;
12      strcpy(ObjectName.StrData, "電源");
13      ObjectName.Length = strlen(ObjectName.StrData);
14
15      ObjectHandle = BD_AddObjectID(ObjectID, &ObjectName); // BI-123 オブジェクトの作成
16      if (ObjectHandle == 0)
17      {
18          /* エラー処理。 */
19      }
20      return 0;
21  }
```

第5章 トラブルシューティングと保証

5.1. トラブル状況と対策

(1) BACnet 接続に関するトラブル

トラブル状況	対処
BACnet デバイスと接続できない。	LAN ケーブルは正しく接続されていますか？ また HUB を介さずに 1 対 1 で接続する場合は、クロスケーブルを使用する必要があります。
	ネットワーク内で IP アドレスが重なっていませんか？ IP アドレスを確認してください。
	BACnet デバイスインスタンス番号が重なっていませんか？ BACnet デバイスインスタンス番号を確認してください。
	本製品の BACnet 規格は正しく設定されていますか？ コンフィギュレーション機能より BACnet 規格を確認してください。
PV が反映されない。または更新されない。	OutOfService プロパティが True になっていませんか？ または Reliability プロパティが True になっていませんか？ これらのプロパティが True になっていると PV プロパティは変化しません。
	プロパティライトを行う際には、Priority を正しく指定しなければいけません。Priority を幾つにして BACnet デバイスからプロパティライトが出されているのかを確認してください。

5.2. 保証について

ご使用に際しましては、以下の製品保証内容をご確認いただきますよう、よろしくお願いいたします。

(1) 無償保証期間と無償保証範囲

無償保証期間中に、製品に当社側の責任による故障や瑕疵（以下併せて「故障」と呼びます）が発生した場合、お買い上げいただきました販売店またはサービス会社を通じて、無償で製品を交換させていただきます。

※出張でのソフトウェア改修（バージョンアップ対応）が必要な場合は、ご相談ください。なお、お客様にて製品の修理や対策を行った場合は、保証の対象外となります。

（弊社からの指定により、お客様にてバージョンアップ対応を行った場合を除きます。）

【無償保証期間】

製品の無償保証期間は、納入後 1 年間とさせていただきます。また、交換品・改修品の無償保証期間は、交換・改修前の無償保証期間を超えて長くなることはありません。

【無償保証範囲】

使用状態、使用方法および使用環境などが、マニュアル、製品本体注意ラベルなどに記載された条件、注意事項などにしたがった正常な状態で使用されている場合に限定させていただきます。

(2) 無償保証適用外事由

無償保証期間内であっても、以下の場合には有償交換とさせていただきます。

- お客様における不適切な保管や取扱い、不注意、過失などにより生じた故障およびお客様のハードウェアまたはソフトウェア設計内容に起因した故障。
- お客様にて当社の了解なく製品に改造などの手を加えたことに起因する故障。
- 当社製品がお客様の機器に組み込まれて使用された場合、お客様の機器が受けている法的規制による安全装置または業界の通念上備えられているべきと判断される機能・構造などを備えていれば回避できたと認められる故障。
- 取扱説明書などに指定された消耗部品が正常に保守・交換されていれば防げたと認められる故障。
- 火災、異常電圧などの不可抗力による外部要因および地震、雷、風水害などの天変地異による故障。
- 当社出荷当時の科学技術の水準では予見できなかった事由による故障。
- その他、当社の責任外の場合またはお客様が当社責任外と認めた故障。

(3) 生産中止後の有償修理期間

当社が有償にて製品の改修を受け付けることができる期間は、その製品の生産中止後 3 年間です。生産中止に関しましては、当社ホームページなどにて報じさせていただきます。

生産中止後の製品供給（補用品も含む）はできません。

(4) 海外でのサービス

海外においては、当社のサポート対象外となります。

(5) 機会損失、二次損失などへの保証責務の除外

無償保証期間の内外を問わず、当社の責に帰することができない事由から生じた損害、当社製品の故障に起因するお客様での機会損失、逸失利益、当社の予見の有無を問わず特別の事情から生じた損害、二次損害、事故補償、当社製品以外への損傷およびその他の業務に対する補償については、当社は責任を負いかねます。

(6) 製品仕様の変更

マニュアルに記載されている仕様は、お断りなしに変更される場合がありますので、あらかじめご承知おき下さい。

(7) 製品の適用について

本製品および他システムに故障・不具合などが発生した場合でも重大な事故にいたらない用途であること、および故障・不具合発生時にはバックアップやフェールセーフ機能が機器外部でシステム的に実施されていることを、ご使用の条件とさせていただきます。

本製品は、一般建物、工業などへの用途を対象とした汎用品として設計・製作されています。したがって、各電力会社殿の原子力発電所およびその他発電所向けなどの公共への影響が大きい用途や、鉄道会社殿および防衛庁殿向けの用途などで、特別品質保証体制をご要求になる用途には、本製品の適用を除外させていただきます。ただし、これらの用途であっても、用途を限定して特別な品質をご要求されないことをお客様にご承認いただいた場合には、適用可能とさせていただきます。

また、航空、医療、鉄道、燃焼・燃料装置、有人搬送装置、娯楽機械、安全機械など人命や財産に大きな影響が予測され、安全面や制御システムに特に高信頼性が要求される用途へのご使用をご検討いただいている場合には、当社窓口へご相談いただき、必要な仕様書の取り交わしなどをさせていただきます。

5.3. サポートについて

本製品は、お買い上げいただきました販売店またはサービス会社を通じ、株式会社ユニテックまでお問い合わせいただけますようお願い申し上げます。

対応時間帯

平日 9:00～12:00 13:00～16:00

（祝祭日、2月19日、および夏季、年末年始など当社の休日は除く）

株式会社ユニテック サポート窓口

〒493-0006 愛知県一宮市木曽川町内割田一の通り 24 番地

E-MAIL によるお問い合わせ

必要事項を記載し、お問い合わせください。

メールアドレス：sales@uni-tec.co.jp

電話によるお問い合わせ

電話番号：0586-87-7819

- ・正確な情報を伝達し合うため、Eメールをご利用いただけますようお願いいたします。
- ・技術的なお問い合わせは、電話での対応をお断りすることがありますので、ご了承ください。

製品に関する情報につきまして

最新情報などは以下の URL からご確認ください。

【URL】 <https://www.uni-tec.co.jp/>

メモ