

DSHEng5 装置／ホスト通信エンジンライブラリ (GEM+GEM300)

ソフトウェア・パッケージ

DSHEng5 GEM 通信エンジン・クラス説明書

Vol - 6

通信メッセージ処理クラス

エンコード / デコード

2019年12月 (改訂-1)

株式会社データマップ

[取り扱い注意]

- この資料ならびにソフトウェアの一部または全部を無断で使用、複製することはできません。
- 本説明書に記述されている内容は予告なしで変更される可能性があります。
- Windows は米国 Microsoft Corporation の登録商標です。
- ユーザーが本ソフトウェアの使用によって生じた遺失履歴、(株) データマップの予見の有無を問わず発生した特別損害、付随的損害、間接損害およびその他の拡大損害に対して責任を負いません。

【改訂履歴】

番号	改訂日付	項目	概略
1.	2019-06-28	初版	
2.	2019.12.16	訂正と記述追加	誤字等の訂正 DSHMSG 構造体のメンバー型を変更した。 txtp, work1, work2 の3つ。 (DSHEng5 が内部処理に使用するメンバー)
2.			
2.			

目 次

1. はじめに.....	1
1. 1 DSHMSG - SECS-II メッセージ構造体.....	2
2. メッセージ ID (S, F) によるエンコード/デコードクラスの説明.....	3
2. 1 class_S1F3 - S1F3/S1F4 エンコード/デコードクラス.....	4
2. 1. 1 EncodeS1F30 - S1F3 のエンコード.....	4
2. 1. 2 DecodeS1F30 - S1F3 のデコード.....	4
2. 1. 3 EncodeS1F40 - S1F4 のエンコード.....	5
2. 1. 4 DecodeS1F40 - S1F4 のデコード.....	5
2. 2 class_S1F11 - S1F11/S1F12 エンコード/デコードクラス.....	6
2. 2. 1 EncodeS1F110 - S1F11 のエンコード.....	6
2. 2. 2 DecodeS1F110 - S1F11 のデコード.....	6
2. 2. 3 EncodeS1F120 - S1F12 のエンコード.....	7
2. 2. 4 DecodeS1F120 - S1F12 のデコード.....	7
2. 3 class_S1F16 エンコード/デコードクラス.....	8
2. 3. 1 EncodeS1F160 - S1F16 のエンコード.....	8
2. 3. 2 DecodeS1F160 - S1F16 のデコード.....	8
2. 4 class_S1F18 エンコード/デコードクラス.....	9
2. 4. 1 EncodeS1F180 - S1F18 のエンコード.....	9
2. 4. 2 DecodeS1F180 - S1F18 のデコード.....	9
2. 5 class_S2F13 - S2F13/S2F14 エンコード/デコードクラス.....	10
2. 5. 1 EncodeS2F130 - S2F13 のエンコード.....	10
2. 5. 2 DecodeS2F130 - S2F13 のデコード.....	10
2. 5. 3 EncodeS2F140 - S2F14 のエンコード.....	11
2. 5. 4 DecodeS2F140 - S2F14 のデコード.....	11
2. 6 class_S2F15 - S2F15/S2F16 エンコード/デコードクラス.....	12
2. 6. 1 EncodeS2F150 - S2F15 のエンコード.....	12
2. 6. 2 DecodeS2F150 - S2F15 のデコード.....	12
2. 6. 3 EncodeS2F160 - S2F16 のエンコード.....	13
2. 6. 4 DecodeS2F160 - S2F16 のデコード.....	13
2. 7 class_S2F17 - S2F17/S2F18 エンコード/デコードクラス.....	14
2. 7. 1 EncodeS2F170 - S2F17 のエンコード.....	14
2. 7. 2 EncodeS2F180 - S2F18 のエンコード.....	14
2. 7. 3 DecodeS2F180 - S2F18 のデコード.....	15
2. 8 class_S2F23 - S2F23/S2F24 エンコード/デコードクラス.....	16
2. 8. 1 EncodeS2F230 - S2F23 のエンコード.....	16
2. 8. 2 DecodeS2F230 - S2F23 のデコード.....	16
2. 8. 3 EncodeS2F240 - S2F24 のエンコード.....	17
2. 8. 4 DecodeS2F240 - S2F24 のデコード.....	17
2. 9 class_S2F29 - S2F29/S2F30 エンコード/デコードクラス.....	18
2. 9. 1 EncodeS2F290 - S2F29 のエンコード.....	18
2. 9. 2 DecodeS2F290 - S2F29 のデコード.....	18
2. 9. 3 EncodeS2F300 - S2F30 のエンコード.....	19
2. 9. 4 DecodeS2F300 - S2F30 のデコード.....	19
2. 10 class_S2F31 - S2F31/S2F32 エンコード/デコードクラス.....	20
2. 10. 1 EncodeS2F310 - S2F31 のエンコード.....	20
2. 10. 2 DecodeS2F310 - S2F31 のデコード.....	20
2. 10. 3 EncodeS2F320 - S2F32 のエンコード.....	21

2. 10. 4	DecodeS2F320 - S2F32 のデコード	21
2. 11	class_S2F33 - S2F33 / S2F34 エンコード/デコードクラス	22
2. 11. 1	EncodeS2F330 - S2F33 のエンコード	22
2. 11. 2	DecodeS2F330 - S2F33 のデコード	22
2. 11. 3	EncodeS2F340 - S2F34 のエンコード	23
2. 11. 4	DecodeS2F340 - S2F34 のデコード	23
2. 12	class_S2F35 - S2F35 / S2F36 エンコード/デコードクラス	24
2. 12. 1	EncodeS2F350 - S2F35 のエンコード	24
2. 12. 2	DecodeS2F350 - S2F35 のデコード	24
2. 12. 3	EncodeS2F360 - S2F36 のエンコード	25
2. 12. 4	DecodeS2F360 - S2F36 のデコード	25
2. 13	class_S2F37 - S2F37 / S2F38 エンコード/デコードクラス	26
2. 13. 1	EncodeS2F370 - S2F37 のエンコード	26
2. 13. 2	DecodeS2F370 - S2F37 のデコード	26
2. 13. 3	EncodeS2F380 - S2F38 のエンコード	27
2. 13. 4	DecodeS2F380 - S2F38 のデコード	27
2. 14	class_S2F41 - S2F41 / S2F42 エンコード/デコードクラス	28
2. 14. 1	EncodeS2F410 - S2F41 のエンコード	28
2. 14. 2	DecodeS2F410 - S2F41 のデコード	28
2. 14. 3	EncodeS2F420 - S2F42 のエンコード	29
2. 14. 4	DecodeS2F420 - S2F42 のデコード	29
2. 15	class_S2F43 - S2F43 / S2F44 エンコード/デコードクラス	30
2. 15. 1	EncodeS2F430 - S2F43 のエンコード	30
2. 15. 2	DecodeS2F430 - S2F43 のデコード	30
2. 15. 3	EncodeS2F440 - S2F44 のエンコード	31
2. 15. 4	DecodeS2F440 - S2F44 のデコード	31
2. 16	class_S2F45 - S2F45 / S2F46 エンコード/デコードクラス	32
2. 16. 1	EncodeS2F450 - S2F45 のエンコード	32
2. 16. 2	DecodeS2F450 - S2F45 のデコード	32
2. 16. 3	EncodeS2F460 - S2F46 のエンコード	33
2. 16. 4	DecodeS2F460 - S2F46 のデコード	33
2. 17	class_S2F47 - S2F47 / S2F48 エンコード/デコードクラス	34
2. 17. 1	EncodeS2F470 - S2F47 のエンコード	34
2. 17. 2	DecodeS2F470 - S2F47 のデコード	34
2. 17. 3	EncodeS2F480 - S2F48 のエンコード	35
2. 17. 4	DecodeS2F480 - S2F48 のデコード	35
2. 18	class_S2F49 - S2F49 / S2F50 エンコード/デコードクラス	36
2. 18. 1	EncodeS2F490 - S2F49 のエンコード	36
2. 18. 2	DecodeS2F490 - S2F49 のデコード	36
2. 18. 3	EncodeS2F500 - S2F50 のエンコード	37
2. 18. 4	DecodeS2F500 - S2F50 のデコード	37
2. 19	class_S3F17 - S3F17 / S3F18 エンコード/デコードクラス	38
2. 19. 1	EncodeS3F170 - S3F17 のエンコード	38
2. 19. 2	DecodeS3F170 - S3F17 のデコード	38
2. 19. 3	EncodeS3F180 - S3F18 のエンコード	39
2. 19. 4	DecodeS3F180 - S3F18 のデコード	39
2. 20	class_S3F23 - S3F23 / S3F24 エンコード/デコードクラス	40
2. 20. 1	EncodeS3F230 - S3F23 のエンコード	40
2. 20. 2	DecodeS3F230 - S3F23 のデコード	40

2. 20. 3	EncodeS3F240 - S3F24 のエンコード	41
2. 20. 4	DecodeS3F240 - S3F24 のデコード	41
2. 21	class_S3F25 - S3F25/S3F26 エンコード/デコードクラス	42
2. 21. 1	EncodeS3F250 - S3F25 のエンコード	42
2. 21. 2	DecodeS3F250 - S3F25 のデコード	42
2. 21. 3	EncodeS3F260 - S3F26 のエンコード	43
2. 21. 4	DecodeS3F260 - S3F26 のデコード	43
2. 22	class_S3F27 - S3F27/S3F28 エンコード/デコードクラス	44
2. 22. 1	EncodeS3F270 - S3F27 のエンコード	44
2. 22. 2	DecodeS3F270 - S3F27 のデコード	44
2. 22. 3	EncodeS3F280 - S3F28 のエンコード	45
2. 22. 4	DecodeS3F280 - S3F28 のデコード	45
2. 23	class_S5F1 - S5F1/S5F2 エンコード/デコードクラス	46
2. 23. 1	EncodeS5F10 - S5F1 のエンコード	46
2. 23. 2	DecodeS5F10 - S5F1 のデコード	47
2. 23. 3	EncodeS5F20 - S5F2 のエンコード	48
2. 23. 4	DecodeS5F20 - S5F2 のデコード	48
2. 24	class_S5F3 - S5F3/S5F4 エンコード/デコードクラス	49
2. 24. 1	EncodeS5F30 - S5F3 のエンコード	49
2. 24. 2	DecodeS5F30 - S5F3 のデコード	49
2. 24. 3	EncodeS5F40 - S5F4 のエンコード	50
2. 24. 4	DecodeS5F40 - S5F4 のデコード	50
2. 25	class_S5F5 - S5F5/S5F6 エンコード/デコードクラス	51
2. 25. 1	EncodeS5F50 - S5F5 のエンコード	51
2. 25. 2	DecodeS5F50 - S5F5 のデコード	51
2. 25. 3	EncodeS5F60 - S5F6 のエンコード	52
2. 25. 4	DecodeS5F60 - S5F6 のデコード	52
2. 26	class_S5F7 - S5F7/S5F8 エンコード/デコードクラス	53
2. 26. 1	EncodeS5F70 - S5F7 のエンコード	53
2. 26. 2	EncodeS5F80 - S5F8 のエンコード	53
2. 26. 3	DecodeS5F80 - S5F8 のデコード	54
2. 27	class_S6F1 - S6F1/S6F2 エンコード/デコードクラス	55
2. 27. 1	EncodeS6F10 - S6F1 のエンコード	55
2. 27. 2	DecodeS6F10 - S6F1 のデコード	55
2. 27. 3	EncodeS6F20 - S6F2 のエンコード	56
2. 27. 4	DecodeS6F20 - S6F2 のデコード	56
2. 28	class_S6F11 - S6F11/S6F12 エンコード/デコードクラス	57
2. 28. 1	EncodeS6F110 - S6F11 のエンコード	57
2. 28. 2	DecodeS6F110 - S6F11 のデコード	57
2. 28. 3	EncodeS6F120 - S6F12 のエンコード	58
2. 28. 4	DecodeS6F120 - S6F12 のデコード	58
2. 29	class_S6F15 - S6F15/S6F16 エンコード/デコードクラス	59
2. 29. 1	EncodeS6F150 - S6F15 のエンコード	59
2. 29. 2	DecodeS6F150 - S6F15 のデコード	59
2. 29. 3	EncodeS6F160 - S6F16 のエンコード	60
2. 29. 4	DecodeS6F160 - S6F16 のデコード	60
2. 30	class_S6F19 - S6F19/S6F20 エンコード/デコードクラス	61
2. 30. 1	EncodeS6F190 - S6F19 のエンコード	61
2. 30. 2	DecodeS6F190 - S6F19 のデコード	61

2. 30. 3	EncodeS6F200 - S6F200 のエンコード	62
2. 30. 4	DecodeS6F200 - S6F200 のデコード	62
2. 31	class_S6F23 - S6F23/S6F24 エンコード/デコードクラス	63
2. 31. 1	EncodeS6F230 - S6F230 のエンコード	63
2. 31. 2	DecodeS6F230 - S6F230 のデコード	63
2. 31. 3	EncodeS6F240 - S6F240 のエンコード	64
2. 31. 4	DecodeS6F240 - S6F240 のデコード	64
2. 32	class_S7F1 - S7F1/S7F2 エンコード/デコードクラス	65
2. 32. 1	EncodeS7F10 - S7F10 のエンコード	65
2. 32. 2	DecodeS7F10 - S7F10 のデコード	65
2. 32. 3	EncodeS7F20 - S7F20 のエンコード	66
2. 32. 4	DecodeS7F20 - S7F20 のデコード	66
2. 33	class_S7F3 - S7F3/S7F4 エンコード/デコードクラス	67
2. 33. 1	EncodeS7F30 - S7F30 のエンコード	67
2. 33. 2	DecodeS7F30 - S7F30 のデコード	67
2. 33. 3	EncodeS7F40 - S7F40 のエンコード	68
2. 33. 4	DecodeS7F40 - S7F40 のデコード	68
2. 34	class_S7F5 - S7F5/S7F6 エンコード/デコードクラス	70
2. 34. 1	EncodeS7F50 - S7F50 のエンコード	70
2. 34. 2	DecodeS7F50 - S7F50 のデコード	70
2. 34. 3	EncodeS7F60 - S7F60 のエンコード	71
2. 34. 4	DecodeS7F60 - S7F60 のデコード	71
2. 35	class_S7F17 - S7F17/S7F18 エンコード/デコードクラス	72
2. 35. 1	EncodeS7F170 - S7F170 のエンコード	72
2. 35. 2	DecodeS7F170 - S7F170 のデコード	72
2. 35. 3	EncodeS7F180 - S7F180 のエンコード	73
2. 35. 4	DecodeS7F180 - S7F180 のデコード	73
2. 36	class_S7F19 - S7F19/S7F20 エンコード/デコードクラス	74
2. 36. 1	EncodeS7F190 - S7F190 のエンコード	74
2. 36. 2	EncodeS7F200 - S7F200 のエンコード	74
2. 36. 3	DecodeS7F200 - S7F200 のデコード	75
2. 37	class_S7F23 - S7F23/S7F24 エンコード/デコードクラス	76
2. 37. 1	EncodeS7F230 - S7F230 のエンコード	76
2. 37. 2	DecodeS7F230 - S7F230 のデコード	76
2. 37. 3	EncodeS7F240 - S7F240 のエンコード	77
2. 37. 4	DecodeS7F240 - S7F240 のデコード	77
2. 38	class_S7F25 - S7F25/S7F26 エンコード/デコードクラス	78
2. 38. 1	EncodeS7F250 - S7F250 のエンコード	78
2. 38. 2	DecodeS7F250 - S7F250 のデコード	78
2. 38. 3	EncodeS7F260 - S7F260 のエンコード	79
2. 38. 4	DecodeS7F260 - S7F260 のデコード	79
2. 39	class_S7F27 - S7F27/S7F28 エンコード/デコードクラス	80
2. 39. 1	EncodeS7F270 - S7F270 のエンコード	80
2. 39. 2	DecodeS7F270 - S7F270 のデコード	80
2. 39. 3	EncodeS7F280 - S7F280 のエンコード	81
2. 39. 4	DecodeS7F280 - S7F280 のデコード	81
2. 40	class_S7F29 - S7F29/S7F30 エンコード/デコードクラス	82
2. 40. 1	EncodeS7F290 - S7F290 のエンコード	82
2. 40. 2	DecodeS7F290 - S7F290 のデコード	82

2. 40. 3	EncodeS7F300 - S7F30 のエンコード.....	83
2. 40. 4	DecodeS7F300 - S7F30 のデコード.....	83
2. 41	class_S10F1 - S10F1/S10F2 エンコード/デコードクラス.....	84
2. 41. 1	EncodeS10F10 - S10F1 のエンコード.....	84
2. 41. 2	DecodeS10F10 - S10F1 のデコード.....	84
2. 41. 3	EncodeS10F20 - S10F2 のエンコード.....	85
2. 41. 4	DecodeS10F20 - S10F2 のデコード.....	85
2. 42	class_S10F3 - S10F3/S10F4 エンコード/デコードクラス.....	86
2. 42. 1	EncodeS10F30 - S10F3 のエンコード.....	86
2. 42. 2	DecodeS10F30 - S10F3 のデコード.....	86
2. 42. 3	EncodeS10F40 - S10F4 のエンコード.....	87
2. 42. 4	DecodeS10F40 - S10F4 のデコード.....	87
2. 43	class_S10F5 - S10F5/S10F6 エンコード/デコードクラス.....	88
2. 43. 1	EncodeS10F50 - S10F5 のエンコード.....	88
2. 43. 2	DecodeS10F50 - S10F5 のデコード.....	88
2. 43. 3	EncodeS10F60 - S10F6 のエンコード.....	89
2. 43. 4	DecodeS10F60 - S10F6 のデコード.....	89
2. 44	class_S14F9 - S14F9/S14F10 エンコード/デコードクラス.....	90
2. 44. 1	EncodeS14F90 - S14F9 のエンコード.....	90
2. 44. 2	DecodeS14F90 - S14F9 のデコード.....	90
2. 44. 3	EncodeS14F100 - S14F10 のエンコード.....	91
2. 44. 4	DecodeS14F100 - S14F10 のデコード.....	91
2. 45	class_S14F11 - S14F11/S14F12 エンコード/デコードクラス.....	92
2. 45. 1	EncodeS14F110 - S14F11 のエンコード.....	92
2. 45. 2	DecodeS14F110 - S14F11 のデコード.....	92
2. 45. 3	EncodeS14F120 - S14F12 のエンコード.....	93
2. 45. 4	DecodeS14F120 - S14F12 のデコード.....	93
2. 46	class_S15F3 - S15F3/S15F4 エンコード/デコードクラス.....	94
2. 46. 1	EncodeS15F30 - S15F3 のエンコード.....	94
2. 46. 2	DecodeS15F30 - S15F3 のデコード.....	94
2. 46. 3	EncodeS15F40 - S15F4 のエンコード.....	95
2. 46. 4	DecodeS15F40 - S15F4 のデコード.....	95
2. 47	class_S15F5 - S15F5/S15F6 エンコード/デコードクラス.....	96
2. 47. 1	EncodeS15F50 - S15F5 のエンコード.....	96
2. 47. 2	DecodeS15F50 - S15F5 のデコード.....	96
2. 47. 3	EncodeS15F60 - S15F6 のエンコード.....	97
2. 47. 4	DecodeS15F60 - S15F6 のデコード.....	97
2. 48	class_S15F7 - S15F7/S15F8 エンコード/デコードクラス.....	98
2. 48. 1	EncodeS15F70 - S15F7 のエンコード.....	98
2. 48. 2	DecodeS15F70 - S15F7 のデコード.....	98
2. 48. 3	EncodeS15F80 - S15F8 のエンコード.....	99
2. 48. 4	DecodeS15F80 - S15F8 のデコード.....	99
2. 49	class_S15F9 - S15F9/S15F10 エンコード/デコードクラス.....	100
2. 49. 1	EncodeS15F90 - S15F9 のエンコード.....	100
2. 49. 2	DecodeS15F90 - S15F9 のデコード.....	100
2. 49. 3	EncodeS15F100 - S15F10 のエンコード.....	101
2. 49. 4	DecodeS15F100 - S15F10 のデコード.....	101
2. 50	class_S15F13 - S15F13/S15F14 エンコード/デコードクラス.....	102
2. 50. 1	EncodeS15F130 - S15F13 のエンコード.....	102

2. 50. 2	DecodeS15F130 - S15F13 のデコード.....	102
2. 50. 3	EncodeS15F140 - S15F14 のエンコード.....	103
2. 50. 4	DecodeS15F140 - S15F14 のデコード.....	103
2. 51	class_S15F17 - S15F17/S15F18 エンコード/デコードクラス.....	104
2. 51. 1	EncodeS15F170 - S15F17 のエンコード.....	104
2. 51. 2	DecodeS15F170 - S15F17 のデコード.....	104
2. 51. 3	EncodeS15F180 - S15F18 のエンコード.....	105
2. 51. 4	DecodeS15F180 - S15F18 のデコード.....	105
2. 52	class_S16F5 - S16F5/S16F6 エンコード/デコードクラス.....	106
2. 52. 1	EncodeS16F50 - S16F5 のエンコード.....	106
2. 52. 2	DecodeS16F50 - S16F5 のデコード.....	106
2. 52. 3	EncodeS16F60 - S16F6 のエンコード.....	107
2. 52. 4	DecodeS16F60 - S16F6 のデコード.....	107
2. 53	class_S16F11 - S16F11/S16F12 エンコード/デコードクラス.....	108
2. 53. 1	EncodeS16F110 - S16F11 のエンコード.....	108
2. 53. 2	DecodeS16F110 - S16F11 のデコード.....	108
2. 53. 3	EncodeS16F120 - S16F12 のエンコード.....	109
2. 53. 4	DecodeS16F120 - S16F12 のデコード.....	109
2. 54	class_S16F15 - S16F15/S16F16 エンコード/デコードクラス.....	110
2. 54. 1	EncodeS16F150 - S16F15 のエンコード.....	110
2. 54. 2	DecodeS16F150 - S16F15 のデコード.....	110
2. 54. 3	EncodeS16F160 - S16F16 のエンコード.....	111
2. 54. 4	DecodeS16F160 - S16F16 のデコード.....	111
2. 55	class_S16F17 - S16F17/S16F18 エンコード/デコードクラス.....	112
2. 55. 1	EncodeS16F170 - S16F17 のエンコード.....	112
2. 55. 2	DecodeS16F170 - S16F17 のデコード.....	112
2. 55. 3	EncodeS16F180 - S16F18 のエンコード.....	113
2. 55. 4	DecodeS16F180 - S16F18 のデコード.....	113
2. 56	class_S16F19 - S16F19/S16F20 エンコード/デコードクラス.....	114
2. 56. 1	EncodeS16F190 - S16F19 のエンコード.....	114
2. 56. 2	EncodeS16F200 - S16F20 のエンコード.....	114
2. 56. 3	DecodeS16F200 - S16F20 のデコード.....	115
2. 57	class_S16F21 - S16F21/S16F22 エンコード/デコードクラス.....	116
2. 57. 1	EncodeS16F210 - S16F21 のエンコード.....	116
2. 57. 2	EncodeS16F220 - S16F22 のエンコード.....	116
2. 57. 3	DecodeS16F220 - S16F22 のデコード.....	117
2. 58	class_S16F27 - S16F27/S16F28 エンコード/デコードクラス.....	118
2. 58. 1	EncodeS16F270 - S16F27 のエンコード.....	118
2. 58. 2	DecodeS16F270 - S16F27 のデコード.....	118
2. 58. 3	EncodeS16F280 - S16F28 のエンコード.....	119
2. 58. 4	DecodeS16F280 - S16F28 のデコード.....	119

1. はじめに

DSHEng5 GEM 通信エンジン説明書は、Vol-1 から 6 までの 6 つの Volume に分けられています。
本説明書の Vol 番号は 6 です。

本説明書では、SECS-II 通信メッセージのエンコード/デコードを行うためのクラスの機能、コンストラクタ、プロパティ、メソッドなどについて説明します。

Vol 番号	文書番号	内容
Vol-1	DSHENG5-19-30321-00	DSHEng5 GEM 通信エンジン説明書 Vol-1 エンジン起動・停止、通信確立関連クラス (EngAPI、GEM 通信確立、予約装置変数関連)
Vol-2	DSHENG5-19-30322-00	変数情報関連クラス (EC, SV, DVVAL, CE, Report, Alarm)
Vol-3	DSHENG5-19-30323-00	プロセス情報関連クラス (PP, FPP, RECIPE, PRJ, CJ, CARRIER, SUBSTRATE)
Vol-4	DSHENG5-19-30324-00	SECS-II メッセージ送信クラス
Vol-5	DSHENG5-19-30325-00	SECS-II 通信メッセージ情報保存クラス
Vol-6	DSHENG5-19-30326-00	SECS-II 通信メッセージエンコード/デコード処理クラス

Vol-6 の各クラスのメソッドは、static (静的) メソッドであり、各クラスにはコンストラクタ、APP が使用するプロパティはありません。

エンコード/デコード処理対象は DSHMSG 構造体と各メッセーが有する情報を保存するクラスということになります。

1. 1 DSHMSG - SECS-II メッセージ構造体

HSMS 通信ドライバーと SECS-II 通信メッセージのやり取りをするための構造体です。
この DSHMSG 構造体は、クラスには属していません。

DSHMSG の構造は以下の通りです。

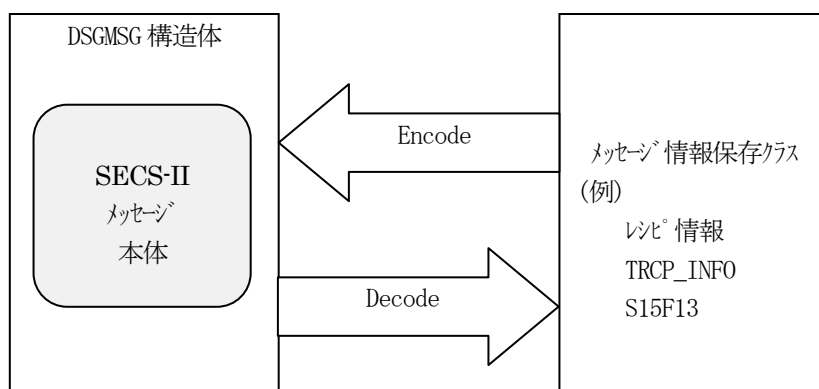
```
public struct DSHMSG
{
    public int stream;           // stream id
    public int function;        // function id
    public int wbit;           // wait bit
    public int length;         // msg length
    public IntPtr buffer;       // msg buffer
    public int error;          // error
    public int next;           // next item
    public IntPtr txtp;         // (内部処理用)
    public int txtc;           // ( " )
    public IntPtr work1;        // ( " )
    public IntPtr work2;        // ( " )
}
```

APP が参照できるメンバーの用途は以下の通りです。

	メンバー	用途
1	stream	stream ID
2	function	function ID
3	wbit	wait bit
4	length	message 長 (byte)
5	buffer	message 格納メモリ
6	next	D_GetItem() で取得する次のデータアイテムの format (HSMS. ICODE_A など)
7	txtp, txtc, work1, work2	内部処理用

2. メッセージ ID (S, F) によるエンコード/デコードクラスの説明

エンコードとデコードの作用は次のようになります。



Encode : メッセージ情報クラスの内容を SECS-II 通信メッセージに符号化します。

Decode : 符号化されているメッセージをメッセージ情報クラスに変換します。

以下、DSHEng5 がサポートする SECS-II メッセージについて、エンコード/デコードするクラスの個別について説明します。

2. 1 class_S1F3 - S1F3/S1F4 エンコード/デコードクラス

2. 1. 1 EncodeS1F3() - S1F3 のエンコード

指定された SVID リストから S1F3 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS1F3(ref DSHMSG smsg, TVID_LIST list)
```

【引数】

smsg
エンコード先のメッセージ保存構造体です。

list
エンコード元の SVID リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list から、引数 smsg に S1F3 メッセージを生成します。

2. 1. 2 DecodeS1F3() - S1F3 のデコード

DSHMSG 構造体に含まれる S1F3 メッセージをデコードして、SVID 配列リストに格納します。

【構文】

```
public static int EncodeS1F3(ref DSHMSG smsg, ref TVID_LIST list)
```

【引数】

smsg
デコード元のメッセージ保存構造体です。

list
デコード先の SVID リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして引数の list に SVID を格納します。

2. 1. 3 EncodeS1F4() - S1F4 のエンコード

指定された SVID リストから S1F4 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS1F4(ref DSHMSG smsg, TVID_LIST list)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

エンコード元の SVID リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。list に未登録の SVID が含まれていた。

【説明】

引数 list から、引数 smsg に S1F4 メッセージを生成します。

SV 情報は、DSHEng5 が自動的に SV 管理情報から取得し、メッセージを生成します。

2. 1. 4 DecodeS1F4() - S1F4 のデコード

DSHMSG 構造体に含まれる S1F4 メッセージをデコードして、SV 情報リストに格納します。

【構文】

```
public static int DecodeS1F4(ref DSHMSG smsg, ref TV_VALUE_LIST list)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

list

デコード先の SV 情報保存リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして引数の list に SV 情報を格納します。

2. 2 class_S1F11 - S1F11/S1F12 エンコード/デコードクラス

2. 2. 1 EncodeS1F11() - S1F11 のエンコード

指定された SVID リストから S1F11 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS1F11(ref DSHMSG smsg, TVID_LIST list)
```

【引数】

smsg
エンコード先のメッセージ保存構造体です。

list
エンコード元の SVID リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list から、引数 smsg に S1F11 メッセージを生成します。

2. 2. 2 DecodeS1F11() - S1F11 のデコード

DSHMSG 構造体に含まれる S1F11 メッセージをデコードして、SVID 配列リストに格納します。

【構文】

```
public static int EncodeS1F11(ref DSHMSG smsg, ref TVID_LIST list)
```

【引数】

smsg
デコード元のメッセージ保存構造体です。

list
デコード先の SVID リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして引数の list に SVID を格納します。

2. 2. 3 EncodeS1F12() - S1F12 のエンコード

指定された SVID リストから S1F12 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS1F12(ref DSHMSG msg, TVID_LIST list)
```

【引数】

msg
エンコードしたメッセージを保存する DSHMSG 構造体です。

list
エンコード元の SVID リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。list に未登録の SVID が含まれていた。

【説明】

引数 list から、引数 msg に S1F12 メッセージを生成します。
SV 名前情報は、DSHEng5 が自動的に SV 管理情報から取得し、メッセージを生成します。

2. 2. 4 DecodeS1F12() - S1F12 のデコード

DSHMSG 構造体に含まれる S1F12 メッセージをデコードして、SV 情報リストに格納します。

【構文】

```
public static int DecodeS1F12(ref DSHMSG msg, ref TSV_NAME_LIST list)
```

【引数】

msg
デコード元のメッセージ保存構造体です。

list
デコード先の SV 名前情報保存リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の list に SV 名前情報を格納します。

2. 3 class_S1F16 エンコード/デコードクラス

2. 3. 1 EncodeS1F16() - S1F16 のエンコード

S1F16 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS1F16(ref DSHMSG msg, int oflack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

oflack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。1

【説明】

引数 oflack から、引数 msg に S1F16 メッセージを生成します。

2. 3. 2 DecodeS1F16() - S1F16 のデコード

DSHMSG 構造体に含まれる S1F16 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS1F16(ref DSHMSG msg, ref int oflack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

oflack

oflack を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の oflack に応答コードを格納します。

2. 4 class_S1F18 エンコード/デコードクラス

2. 4. 1 EncodeS1F18() - S1F18のエンコード

S1F18メッセージをDSHMSG構造体の中に生成します。

【構文】

```
public static int EncodeS1F18(ref DSHMSG msg, int onlack)
```

【引数】

msg

エンコードしたメッセージを保存するDSHMSG構造体です。

onlack

応答ACKです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。1

【説明】

引数 onlack から、引数 msg にS1F18メッセージを生成します。

2. 4. 2 DecodeS1F18() - S1F18のデコード

DSHMSG構造体に含まれるS1F18メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS1F18(ref DSHMSG msg, ref int onlack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

onlack

onlack を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の onlack に応答コードを格納します。

2. 5 class_S2F13 - S2F13/S2F14 エンコード/デコードクラス

2. 5. 1 EncodeS2F13() - S2F13 のエンコード

指定された ECID リストから S2F13 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F13(ref DSHMSG msg, TVID_LIST list)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

list

エンコード元の ECID リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list から、引数 msg に S2F13 メッセージを生成します。

2. 5. 2 DecodeS2F13() - S2F13 のデコード

DSHMSG 構造体に含まれる S2F13 メッセージをデコードして、ECID 配列リストに格納します。

【構文】

```
public static int DecodeS2F13(ref DSHMSG msg, ref TVID_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

デコード先の ECID リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の list に ECID を格納します。

2. 5. 3 EncodeS2F14() - S2F14 のエンコード

指定された ECID リストから S2F14 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F14(ref DSHMSG msg, TVID_LIST list)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

エンコード元の ECID リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。list に未登録の ECID が含まれていた。

【説明】

引数 list から、引数 msg に S2F14 メッセージを生成します。

EC 情報は、DSHEng5 が自動的に EC 管理情報から取得し、メッセージを生成します。

2. 5. 4 DecodeS2F14() - S2F14 のデコード

DSHMSG 構造体に含まれる S2F14 メッセージをデコードして、EC 情報リストに格納します。

【構文】

```
public static int DecodeS2F14(ref DSHMSG msg, ref TV_VALUE_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

デコード先の EC 情報保存リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の list に EC 情報を格納します。

2. 6 class_S2F15 - S2F15/S2F16 エンコード/デコードクラス

2. 6. 1 EncodeS2F15() - S2F15 のエンコード

指定された ECID と EC 値リストから S2F15 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F15(ref DSHMSG smsg, TVID_VAL_LIST list)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

list

エンコード元の ECID とその値のリストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list から、引数 smsg に S2F15 メッセージを生成します。

2. 6. 2 DecodeS2F15() - S2F15 のデコード

DSHMSG 構造体に含まれる S2F15 メッセージをデコードして、ECID 配列リストに格納します。

【構文】

```
public static int DecodeS2F15(ref DSHMSG smsg, ref TVID_VAL_LIST list)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

list

デコード先の ECID とその値リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして引数の list に ECID とその値を格納します。

2. 6. 3 EncodeS2F16() - S2F16 のエンコード

指定された ECID リストから S2F16 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F16(ref DSHMSG msg, int ack)
```

【引数】

msg
エンコードしたメッセージを保存する DSHMSG 構造体です。

ack
応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S2F16 メッセージを生成します。

2. 6. 4 DecodeS2F16() - S2F16 のデコード

DSHMSG 構造体に含まれる S2F16 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS2F16( ref DSHMSG msg, ref int ack)
```

【引数】

msg
デコード元のメッセージ保存構造体です。

ack
応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 7 class_S2F17 - S2F17/S2F18 エンコード/デコードクラス

2. 7. 1 EncodeS2F17() - S2F17 のエンコード

S2F17 メッセージを DSHMSG 構造体の中に生成します。(Header だけです)

【構文】

```
public static int EncodeS2F17(ref DSHMSG msg)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 msg に S2F17 メッセージを生成します。

2. 7. 2 EncodeS2F18() - S2F18 のエンコード

現在の日付時刻から S2F18 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F18(ref DSHMSG msg)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

現在日付時刻から、引数 msg に S1F18 メッセージを生成します。

日付時刻情報は、DSHEng5 が Windows から取得します。

2. 7. 3 DecodeS2F18() - S2F18 のデコード

DSHMSG 構造体の S2F18 メッセージから日付時刻データを取得します。

【構文】

```
public static int EncodeS2F18(ref DSHMSG smsg, ref string time_data)
```

【引数】

smsg

デコード対象のメッセージを保存する DSHMSG 構造体です。

time_data

日付時刻を格納するための文字列です。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Dencode できなかった。

【説明】

現在日付時刻から、引数 smsg に S1F18 含まれる日付時刻データを引数 time_data に設定します。

2. 8 class_S2F23 - S2F23/S2F24 エンコード/デコードクラス

2. 8. 1 EncodeS2F23() - S2F23 のエンコード

トレース情報から S2F23 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F23( ref DSHMSG msg, TTRACE_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

トレース情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info の情報から、引数 msg に S2F23 メッセージを生成します。

2. 8. 2 DecodeS2F23() - S2F23 のデコード

DSHMSG 構造体に含まれる S2F23 メッセージをデコードして、トレース情報を info に格納します。

【構文】

```
public static int DecodeS2F23( ref DSHMSG msg, ref TTRACE_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

トレース情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F23 メッセージをデコードして、トレース情報を info に格納します。

2. 8. 3 EncodeS2F24() - S2F24 のエンコード

引数の ACK 情報から S2F24 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F24(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S2F24 メッセージを生成します。

2. 8. 4 DecodeS2F24() - S2F24 のデコード

DSHMSG 構造体に含まれる S2F24 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS2F24( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 9 class_S2F29 - S2F29/S2F30 エンコード/デコードクラス

2. 9. 1 EncodeS2F29() - S2F29 のエンコード

ECID リストから S2F29 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F29( ref DSHMSG msg, TVID_LIST list)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

list

ECID が格納されているリストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list の情報から、引数 msg に S2F29 メッセージを生成します。

2. 9. 2 DecodeS2F29() - S2F29 のデコード

DSHMSG 構造体に含まれる S2F29 メッセージをデコードして、ECID を list に格納します。

【構文】

```
public static int DecodeS2F29( ref DSHMSG msg, ref TVID_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

ECID を格納するリストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F29 メッセージをデコードして、ECLID リスト情報を list に格納します。

2. 9. 3 EncodeS2F30() - S2F30 のエンコード

指定された ECID リストから S2F30 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F30(ref DSHMSG smsg, TVID_LIST list)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

ECID が格納されているリストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 list に含まれる ECID の名前情報を取得し、引数 smsg に S2F30 メッセージを生成します。

2. 9. 4 DecodeS2F30() - S2F30 のデコード

DSHMSG 構造体に含まれる S2F30 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS2F30(ref DSHMSG smsg, ref TEC_NAME_LIST list)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

list

EC 名前情報が格納されている配列リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして、引数の list に EC 名前情報を格納します。

2. 10 class_S2F31 - S2F31/S2F32 エンコード/デコードクラス

2. 10. 1 EncodeS2F31 () - S2F31 のエンコード

日付時刻データから S2F31 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F31( ref DSHMSG msg, string date_time)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

date_time

日付時刻データ。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(date_time=null)

【説明】

引数 date_time の情報から、引数 msg に S2F31 メッセージを生成します。

2. 10. 2 DecodeS2F31 () - S2F31 のデコード

DSHMSG 構造体に含まれる S2F31 メッセージをデコードして、日付時刻データ情報を date_time に格納します。

【構文】

```
public static int DecodeS2F31( ref DSHMSG msg, ref string date_time)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

date_time

日付時刻情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F31 メッセージをデコードして、日付時刻情報を date_time に格納します。

2. 10. 3 EncodeS2F32() - S2F32 のエンコード

与えられた ACK から S2F32 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F32(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S2F32 メッセージを生成します。

2. 10. 4 DecodeS2F32() - S2F32 のデコード

DSHMSG 構造体に含まれる S2F32 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS2F32( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 11 class_S2F33 - S2F33/S2F34 エンコード/デコードクラス

2. 11. 1 EncodeS2F33() - S2F33 のエンコード

レポートリンク情報から S2F33 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F33( ref DSHMSG msg, TRP_LIST list)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

list

レポートリンク情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list のレポートリンクリスト情報から、引数 msg に S2F33 メッセージを生成します。

2. 11. 2 DecodeS2F33() - S2F33 のデコード

DSHMSG 構造体に含まれる S2F33 メッセージをデコードして、レポートリンク情報を list に格納します。

【構文】

```
public static int DecodeS2F33( ref DSHMSG msg, ref TRP_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

レポートリンク情報リストを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F33 メッセージをデコードして、レポートリンク情報を list に格納します。

2. 11. 3 EncodeS2F34() - S2F34 のエンコード

与えられた ACK から S2F34 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F34(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S2F34 メッセージを生成します。

2. 11. 4 DecodeS2F34() - S2F34 のデコード

DSHMSG 構造体に含まれる S2F34 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS2F34( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 12 class_S2F35 - S2F35/S2F36 エンコード/デコードクラス

2. 12. 1 EncodeS2F35 () - S2F35 のエンコード

CE リンク情報から S2F35 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F35( ref DSHMSG msg, TCE_LIST list)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

list

CE リンク情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list の CE リンクリスト情報から、引数 msg に S2F35 メッセージを生成します。

2. 12. 2 DecodeS2F35 () - S2F35 のデコード

DSHMSG 構造体に含まれる S2F35 メッセージをデコードして、CE リンク情報を list に格納します。

【構文】

```
public static int DecodeS2F35( ref DSHMSG msg, ref TCE_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

CE リンク情報リストを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F35 メッセージをデコードして、CE リンク情報を list に格納します。

2. 12. 3 EncodeS2F36() - S2F36 のエンコード

与えられた ACK から S2F36 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F36(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S2F36 メッセージを生成します。

2. 12. 4 DecodeS2F36() - S2F36 のデコード

DSHMSG 構造体に含まれる S2F36 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS2F36(ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 13 class_S2F37 - S2F37/S2F38 エンコード/デコードクラス

2. 13. 1 EncodeS2F37() - S2F37 のエンコード

CEED と CEID リスト情報から S2F37 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F37( ref DSHMSG msg, TEDER_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

CEED と CEID リスト情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info の CEED と CEID リスト情報から、引数 msg に S2F37 メッセージを生成します。

2. 13. 2 DecodeS2F37() - S2F37 のデコード

DSHMSG 構造体に含まれる S2F37 メッセージをデコードして、CEED と CEID リスト情報を info に格納します。

【構文】

```
public static int DecodeS2F37( ref DSHMSG msg, ref TEDER_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

CEED と CEID リストを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F37 メッセージをデコードして、CE 情報を info に格納します。

2. 13. 3 EncodeS2F38() - S2F38 のエンコード

与えられた ACK から S2F38 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F38(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S2F38 メッセージを生成します。

2. 13. 4 DecodeS2F38() - S2F38 のデコード

DSHMSG 構造体に含まれる S2F38 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS2F38( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 14 class_S2F41 - S2F41/S2F42 エンコード/デコードクラス

2. 14. 1 EncodeS2F41 () - S2F41 のエンコード

ホストコマンド情報から S2F41 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F41( ref DSHMSG msg, TRCMD_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

ホストコマンド情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info のホストコマンド情報から、引数 msg に S2F41 メッセージを生成します。

2. 14. 2 DecodeS2F41 () - S2F41 のデコード

DSHMSG 構造体に含まれる S2F41 メッセージをデコードして、ホストコマンド情報を info に格納します。

【構文】

```
public static int DecodeS2F41( ref DSHMSG msg, ref TRCMD_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

ホストコマンド情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F41 メッセージをデコードして、ホストコマンド情報を info に格納します。

2. 14. 3 EncodeS2F42() - S2F42 のエンコード

与えられた応答情報から S2F42 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F42(ref DSHMSG msg, TRCMD_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S2F42 メッセージを生成します。

2. 14. 4 DecodeS2F42() - S2F42 のデコード

DSHMSG 構造体に含まれる S2F42 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS2F42(ref DSHMSG msg, ref TRCMD_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsp_info に応答情報を格納します。

2. 15 class_S2F43 - S2F43/S2F44 エンコード/デコードクラス

2. 15. 1 EncodeS2F43 () - S2F43 のエンコード

スプール情報から S2F43 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F43( ref DSHMSG msg, TSPPOOL_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

スプール情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info のスプール情報から、引数 msg に S2F43 メッセージを生成します。

2. 15. 2 DecodeS2F43 () - S2F43 のデコード

DSHMSG 構造体に含まれる S2F43 メッセージをデコードして、スプール情報を info に格納します。

【構文】

```
public static int DecodeS2F43( ref DSHMSG msg, ref TSPPOOL_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

スプールを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F43 メッセージをデコードして、スプール情報を info に格納します。

2. 15. 3 EncodeS2F44() - S2F44 のエンコード

与えられた応答情報から S2F44 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F44(ref DSHMSG msg, TSPool_Err_Info rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S2F44 メッセージを生成します。

2. 15. 4 DecodeS2F44() - S2F44 のデコード

DSHMSG 構造体に含まれる S2F44 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS2F44(ref DSHMSG msg, ref TSPool_Err_Info rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsp_info に応答情報を格納します。

2. 16 class_S2F45 - S2F45/S2F46 エンコード/デコードクラス

2. 16. 1 EncodeS2F45() - S2F45 のエンコード

リミット情報リストから S2F45 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F45( ref DSHMSG msg, TLIMIT_LIST lmt_list)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

lmt_list

リミット情報リストが保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(lmt_list=null)

【説明】

引数 lmt_list のリミット情報リストから、引数 msg に S2F45 メッセージを生成します。

2. 16. 2 DecodeS2F45() - S2F45 のデコード

DSHMSG 構造体に含まれる S2F45 メッセージをデコードして、リミット情報リストを lmt_list に格納します。

【構文】

```
public static int DecodeS2F45( ref DSHMSG msg, ref TLIMIT_LIST lmt_list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

lmt_list

リミット情報リストを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F45 メッセージをデコードして、リミット情報リストを lmt_list に格納します。

2. 16. 3 EncodeS2F46() - S2F46 のエンコード

与えられた応答情報から S2F46 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F46(ref DSHMSG msg, TLIMIT_ERR_LIST list)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 list から、引数 msg に S2F46 メッセージを生成します。

2. 16. 4 DecodeS2F46() - S2F46 のデコード

DSHMSG 構造体に含まれる S2F46 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS2F46(ref DSHMSG msg, ref TLIMIT_ERR_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の list に応答情報を格納します。

2. 17 class_S2F47 - S2F47/S2F48 エンコード/デコードクラス

2. 17. 1 EncodeS2F47() - S2F47 のエンコード

変数 ID リストから S2F47 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F47(ref DSHMSG msg, TVID_LIST vid_list)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

vid_list

変数 ID リストが保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(vid_list=null)

【説明】

引数 vid_list の変数 ID リストから、引数 msg に S2F47 メッセージを生成します。

2. 17. 2 DecodeS2F47() - S2F47 のデコード

DSHMSG 構造体に含まれる S2F47 メッセージをデコードして、変数 ID を vid_list に格納します。

【構文】

```
public static int DecodeS2F47(ref DSHMSG msg, ref TVID_LIST vid_list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

vid_list

変数リストを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F47 メッセージをデコードして、変数 ID を vid_list に格納します。

2. 17. 3 EncodeS2F48() - S2F48 のエンコード

与えられた応答情報から S2F48 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F48(ref DSHMSG msg, TLIMIT_RSP_LIST list)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 list から、引数 msg に S2F48 メッセージを生成します。

2. 17. 4 DecodeS2F48() - S2F48 のデコード

DSHMSG 構造体に含まれる S2F48 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS2F48(ref DSHMSG msg, ref TLIMIT_RSP_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の list に応答情報を格納します。

2. 18 class_S2F49 - S2F49/S2F50 エンコード/デコードクラス

2. 18. 1 EncodeS2F49 () - S2F49 のエンコード

拡張リモートコマンド情報から S2F49 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F49( ref DSHMSG msg, TERCMD_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

拡張リモートコマンド情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info の拡張リモートコマンド情報から、引数 msg に S2F49 メッセージを生成します。

2. 18. 2 DecodeS2F49 () - S2F49 のデコード

DSHMSG 構造体に含まれる S2F49 メッセージをデコードして、拡張リモートコマンド情報を info に格納します。

【構文】

```
public static int DecodeS2F49( ref DSHMSG msg, ref TERCMD_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

拡張リモートコマンド情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S2F49 メッセージをデコードして、拡張リモートコマンド情報を info に格納します。

2. 18. 3 EncodeS2F50() - S2F50 のエンコード

与えられた応答情報から S2F50 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS2F50(ref DSHMSG msg, TRCMD_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S2F50 メッセージを生成します。

2. 18. 4 DecodeS2F50() - S2F50 のデコード

DSHMSG 構造体に含まれる S2F50 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS2F50(ref DSHMSG msg, ref TRCMD_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsp_info に応答情報を格納します。

2. 19 class_S3F17 - S3F17/S3F18 エンコード/デコードクラス

2. 19. 1 EncodeS3F17() - S3F17のエンコード

キャリアアクション情報から S3F17 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F17( ref DSHMSG msg, TCACT_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

キャリアアクション情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info のキャリアアクション情報から、引数 msg に S3F17 メッセージを生成します。

2. 19. 2 DecodeS3F17() - S3F17のデコード

DSHMSG 構造体に含まれる S3F17 メッセージをデコードして、キャリアアクション情報を info に格納します。

【構文】

```
public static int DecodeS3F17( ref DSHMSG msg, ref TCACT_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

キャリアアクション情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S3F17 メッセージをデコードして、キャリアアクション情報を info に格納します。

2. 19. 3 EncodeS3F18() - S3F18 のエンコード

与えられた応答情報から S3F18 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F18(ref DSHMSG msg, TCACT_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S3F18 メッセージを生成します。

2. 19. 4 DecodeS3F18() - S3F18 のデコード

DSHMSG 構造体に含まれる S3F18 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS3F18(ref DSHMSG msg, ref TCACT_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsp_info に応答情報を格納します。

2. 20 class_S3F23 - S3F23/S3F24 エンコード/デコードクラス

2. 20. 1 EncodeS3F23() - S3F23 のエンコード

ポートグループアクション情報から S3F23 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F23( ref DSHMSG msg, TPORTG_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

ポートグループアクション情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info のポートグループアクション情報から、引数 msg に S3F23 メッセージを生成します。

2. 20. 2 DecodeS3F23() - S3F23 のデコード

DSHMSG 構造体に含まれる S3F23 メッセージをデコードして、ポートグループアクション情報を info に格納します。

【構文】

```
public static int DecodeS3F23( ref DSHMSG msg, ref TPORTG_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

ポートグループアクション情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S3F23 メッセージをデコードして、ポートグループアクション情報を info に格納します。

2. 20. 3 EncodeS3F24() - S3F24 のエンコード

与えられた応答情報から S3F24 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F24(ref DSHMSG msg, TCACT_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S3F24 メッセージを生成します。

2. 20. 4 DecodeS3F24() - S3F24 のデコード

DSHMSG 構造体に含まれる S3F24 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS3F24(ref DSHMSG msg, ref TCACT_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsp_info に応答情報を格納します。

2. 21 class_S3F25 - S3F25/S3F26 エンコード/デコードクラス

2. 21. 1 EncodeS3F25() - S3F25 のエンコード

ポートアクション情報から S3F25 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F25( ref DSHMSG msg, TPORT_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

ポートアクション情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info のポートアクション情報から、引数 msg に S3F25 メッセージを生成します。

2. 21. 2 DecodeS3F25() - S3F25 のデコード

DSHMSG 構造体に含まれる S3F25 メッセージをデコードして、ポートアクション情報を info に格納します。

【構文】

```
public static int DecodeS3F25( ref DSHMSG msg, ref TPORT_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

ポートアクション情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S3F25 メッセージをデコードして、ポートアクション情報を info に格納します。

2. 21. 3 EncodeS3F26() - S3F26 のエンコード

与えられた応答情報から S3F26 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F26(ref DSHMSG msg, TCACT_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S3F26 メッセージを生成します。

2. 21. 4 DecodeS3F26() - S3F26 のデコード

DSHMSG 構造体に含まれる S3F26 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS3F26(ref DSHMSG msg, ref TCACT_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsp_info に応答情報を格納します。

2. 22 class_S3F27 - S3F27/S3F28 エンコード/デコードクラス

2. 22. 1 EncodeS3F27() - S3F27 のエンコード

アクセス変更情報から S3F27 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F27( ref DSHMSG msg, TACCESS_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

アクセス変更情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info のアクセス変更情報から、引数 msg に S3F27 メッセージを生成します。

2. 22. 2 DecodeS3F27() - S3F27 のデコード

DSHMSG 構造体に含まれる S3F27 メッセージをデコードして、アクセス変更情報を info に格納します。

【構文】

```
public static int DecodeS3F27( ref DSHMSG msg, ref TACCESS_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

アクセス変更情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S3F27 メッセージをデコードして、アクセス変更情報を info に格納します。

2. 22. 3 EncodeS3F28() - S3F28 のエンコード

与えられた応答情報から S3F28 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS3F28(ref DSHMSG msg, TCACT_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S3F28 メッセージを生成します。

2. 22. 4 DecodeS3F28() - S3F28 のデコード

DSHMSG 構造体に含まれる S3F28 メッセージをデコードし、応答情報を取得します。

【構文】

```
public static int DecodeS3F28(ref DSHMSG msg, ref TCACT_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

応答情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsp_info に応答情報を格納します。

2. 23 class_S5F1 - S5F1/S5F2 エンコード/デコードクラス

2. 23. 1 EncodeS5F1() - S5F1 のエンコード

アラーム ID から S5F1 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F1( ref DSHMSG smsg, UInt32 id, int on_off )
```

【引数】

smsg エンコード先のメッセージ保存構造体です。
 id アラーム ID です。
 on_off 発生/復旧(1/0)フラグです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 id と on_off から、アラーム情報を取得し、引数 smsg に S5F1 メッセージを生成します。

2. 23. 2 DecodeS5F1() - S5F1 のデコード

DSHMSG 構造体に含まれる S5F1 メッセージをデコードして、アラーム情報とアラーム発生/復旧フラグを取得します。

【構文】

```
public static int DecodeS5F1(ref DSHMSG msg, ref TAL_INFO info, ref int on_off)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

アラーム情報を保存します。

on_off

発生/復旧(1/0)フラグを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S5F1 メッセージをデコードして、アラーム情報と発生/復旧フラグを取得します。

2. 23. 3 EncodeS5F2() - S5F2 のエンコード

与えられた ACK から S5F2 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F2(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S5F2 メッセージを生成します。

2. 23. 4 DecodeS5F2() - S5F2 のデコード

DSHMSG 構造体に含まれる S5F2 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS5F2(ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 24 class_S5F3 - S5F3/S5F4 エンコード/デコードクラス

2. 24. 1 EncodeS5F3() - S5F3 のエンコード

TALS5F3_INFO のインスタンス情報 info から S5F3 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F3( ref DSHMSG smsg, TAL_S5F3_INFO info)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

info

TAL_S5F3_INFO クラス(アラーム ID リストと有効/無効指定フラグ)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 info から引数 smsg に S5F3 メッセージを生成します。

2. 24. 2 DecodeS5F3() - S5F3 のデコード

DSHMSG 構造体に含まれる S5F3 メッセージをデコードしてアラーム ID リストと発生/復旧フラグ情報を取得します。

【構文】

```
public static int DecodeS5F3(ref DSHMSG smsg, ref TAL_S5F3_INFO info)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

info

デコード結果を保存する TAL_S5F3_INFO クラス(アラーム ID リストと有効/無効指定フラグ)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg 含まれる S5F3 メッセージをデコードして、アラーム有効/無効情報を引数 info に取得します。

2. 24. 3 EncodeS5F4() - S5F4 のエンコード

与えられた ACK から S5F4 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F4(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S5F4 メッセージを生成します。

2. 24. 4 DecodeS5F4() - S5F4 のデコード

DSHMSG 構造体に含まれる S5F4 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS5F4( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 25 class_S5F5 - S5F5/S5F6 エンコード/デコードクラス

2. 25. 1 EncodeS5F5 () - S5F5 のエンコード

TALS5F5_INFO のインスタンス情報 info から S5F5 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F5( ref DSHMSG msg, TAL_S5F5_INFO info )
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TAL_S5F5_INFO クラス(アラーム ID リスト)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 info から引数 msg に S5F5 メッセージを生成します。

2. 25. 2 DecodeS5F5 () - S5F5 のデコード

DSHMSG 構造体に含まれる S5F5 メッセージをデコードしてアラーム情報を取得します。

【構文】

```
public static int DecodeS5F5(ref DSHMSG msg, ref TAL_S5F5_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

デコード結果を保存する TAL_S5F5_INFO クラス(アラーム ID リスト)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S5F5 メッセージをデコードして、アラーム ID リストを引数 info に取得します。

2. 25. 3 EncodeS5F6() - S5F6 のエンコード

与えられた ACK から S5F6 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F6( ref DSHMSG msg, TAL_S5F6_LIST list)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

アラーム情報リストです。(ALCD, ALID, ALTX)

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 list から、引数 msg に S5F6 メッセージを生成します。

2. 25. 4 DecodeS5F6() - S5F6 のデコード

DSHMSG 構造体に含まれる S5F6 メッセージをデコードし、TAL_S5F6_LIST のインスタンスに取得します。

【構文】

```
public static int DecodeS5F6( ref DSHMSG msg, ref TAL_S5F6_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

アラーム情報を保存するための TAL_S5F6_LIST のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてアラーム情報を引数 list に保存します。

2. 26 class_S5F7 - S5F7/S5F8 エンコード/デコードクラス

2. 26. 1 EncodeS5F7() - S5F7 のエンコード

S5F7 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F7( ref DSHMSG smsg)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

smsg に S5F7 メッセージを生成します。(header だけです)

2. 26. 2 EncodeS5F8() - S5F8 のエンコード

与えられた ACK から S5F8 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS5F8( ref DSHMSG smsg, TAL_S5F6_LIST list)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

アラーム情報リストです。(ALCD, ALID, ALTX)

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 list から、引数 smsg に S5F8 メッセージを生成します。

2. 26. 3 DecodeS5F8() - S5F8 のデコード

DSHMSG 構造体に含まれる S5F8 メッセージをデコードし、TAL_S5F6_LIST のインスタンスに取得します。

【構文】

```
public static int DecodeS5F8( ref DSHMSG smsg, ref TAL_S5F6_LIST list)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

list

アラーム情報を保存するための TAL_S5F6_LIST のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードしてアラーム情報を引数 list に保存します。

2. 27 class_S6F1 - S6F1/S6F2 エンコード/デコードクラス

2. 27. 1 EncodeS6F1() - S6F1 のエンコード

TALS6F1_INFO のインスタンス情報 minfo から S6F1 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F1( ref DSHMSG msg, TTRACE_MON minfo)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

minfo

TTRACE_MON クラス(トレース結果データ保存)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 minfo から引数 msg に S6F1 メッセージを生成します。

2. 27. 2 DecodeS6F1() - S6F1 のデコード

DSHMSG 構造体に含まれる S6F1 メッセージをデコードしてトレース結果情報を取得します。

【構文】

```
public static int DecodeS6F1(ref DSHMSG msg, ref TTRACE_MON minfo)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

minfo

デコード結果を保存する TTRACE_MON クラス(アラーム ID リストと有効/無効指定フラグ)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S6F1 メッセージをデコードして、トレース結果情報を引数 minfo に取得します。

2. 27. 3 EncodeS6F2() - S6F2 のエンコード

与えられた ACK から S6F2 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F2(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S6F2 メッセージを生成します。

2. 27. 4 DecodeS6F2() - S6F2 のデコード

DSHMSG 構造体に含まれる S6F2 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS6F2( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 28 class_S6F11 - S6F11/S6F12 エンコード/デコードクラス

2. 28. 1 EncodeS6F11() - S6F11 のエンコード

CEID から S6F11 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F11( ref DSHMSG msg, UInt32 ceid)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

ceid

CEID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 ceid から、CE 情報を取得し、引数 msg に S6F11 メッセージを生成します。

2. 28. 2 DecodeS6F11() - S6F11 のデコード

DSHMSG 構造体に含まれる S6F11 メッセージをデコードして、CE 情報を取得します。

【構文】

```
public static int DecodeS6F11(ref DSHMSG msg, ref TCE_CONTENT info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

CE 情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S6F11 メッセージをデコードして、CE 情報を引数 info に取得します。

2. 28. 3 EncodeS6F12() - S6F12 のエンコード

与えられた ACK から S6F12 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F12(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S6F12 メッセージを生成します。

2. 28. 4 DecodeS6F12() - S6F12 のデコード

DSHMSG 構造体に含まれる S6F12 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS6F12( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 29 class_S6F15 - S6F15/S6F16 エンコード/デコードクラス

2. 29. 1 EncodeS6F15() - S6F15 のエンコード

CEID から S6F15 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F15( ref DSHMSG msg, UInt32 ceid)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

ceid

CEID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 ceid から、CE 情報を取得し、引数 msg に S6F15 メッセージを生成します。

2. 29. 2 DecodeS6F15() - S6F15 のデコード

DSHMSG 構造体に含まれる S6F15 メッセージをデコードして、CEID を取得します。

【構文】

```
public static int DecodeS6F15(ref DSHMSG msg, ref UInt32 ceid)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ceid

取得した CEID を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S6F15 メッセージをデコードして、CEID を引数 ceid に格納します。

2. 29. 3 EncodeS6F16() - S6F16 のエンコード

指定された CEID の CE 情報から S6F16 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F16( ref DSHMSG msg, UInt32 ceid)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ceid

CEID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ceid から CE 情報を取り出し、引数 msg に S6F16 メッセージを生成します。

2. 29. 4 DecodeS6F16() - S6F16 のデコード

DSHMSG 構造体に含まれる S6F16 メッセージをデコードし、CE 情報を取得します。

【構文】

```
public static int DecodeS6F16(ref DSHMSG msg, ref TCE_CONTENT t_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

t_info

メッセージから得られた CE 情報を保存する TCE_CONTENT のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の t_info、TCE_CONTENT クラスのインスタンスに格納します。

2. 30 class_S6F19 - S6F19/S6F20 エンコード/デコードクラス

2. 30. 1 EncodeS6F19() - S6F19 のエンコード

レポート ID から S6F19 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F19( ref DSHMSG msg, UInt32 rpid)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

rpid

レポート ID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 rpid から、レポート情報を取得し、引数 msg に S6F19 メッセージを生成します。

2. 30. 2 DecodeS6F19() - S6F19 のデコード

DSHMSG 構造体に含まれる S6F19 メッセージをデコードして、レポート ID を取得します。

【構文】

```
public static int DecodeS6F19(ref DSHMSG msg, ref UInt32 rpid)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rpid

レポート ID を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S6F19 メッセージをデコードして、レポート ID を引数 rpid に格納します。

2. 30. 3 EncodeS6F20() - S6F20 のエンコード

指定されたレポート ID のレポート情報から S6F20 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F20( ref DSHMSG msg, UInt32 rpid)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rpid

レポート ID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rpid からレポート情報を取り出し、引数 msg に S6F20 メッセージを生成します。

2. 30. 4 DecodeS6F20() - S6F20 のデコード

DSHMSG 構造体に含まれる S6F20 メッセージをデコードし、レポート情報を取得します。

【構文】

```
public static int DecodeS6F20(ref DSHMSG msg, ref TRP_CONTENT t_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

t_info

メッセージから得られたレポート情報を保存する TRP_CONTENT のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の t_info、TRP_CONTENT クラスのインスタンスに格納します。

2. 31 class_S6F23 - S6F23/S6F24 エンコード/デコードクラス

2. 31. 1 EncodeS6F23() - S6F23 のエンコード

スプール要求コードから S6F23 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F23( ref DSHMSG msg, int rsdc)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

rsdc

スプール要求コードです。(0=スプールメッセージの転送、1=スプールメッセージの破棄)

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 rsdc から引数 msg に S6F23 メッセージを生成します。

2. 31. 2 DecodeS6F23() - S6F23 のデコード

DSHMSG 構造体に含まれる S6F23 メッセージをデコードして、スプール要求コードを取得します。

【構文】

```
public static int DecodeS6F23(ref DSHMSG msg, ref int rsdc)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsdc

スプール要求コードを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S6F23 メッセージをデコードして、スプール要求コードを引数 rsdc に格納します。

2. 31. 3 EncodeS6F24() - S6F24 のエンコード

指定されたスプールメッセージ要求確認 RSDA から S6F23 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS6F24( ref DSHMSG msg, UInt32 rsda)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsda

スプールメッセージ要求確認コード

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsda からレポート情報を取り出し、引数 msg に S6F24 メッセージを生成します。

2. 31. 4 DecodeS6F24() - S6F24 のデコード

DSHMSG 構造体に含まれる S6F24 メッセージをデコードし、スプールメッセージ要求確認コードを取得します。

【構文】

```
public static int DecodeS6F24(ref DSHMSG msg, ref int rsda)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsda

スプールメッセージ要求確認コード格納メモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の rsda にスプールメッセージ要求確認コードを格納します。

2. 32 class_S7F1 - S7F1/S7F2 エンコード/デコードクラス

2. 32. 1 EncodeS7F1() - S7F1 のエンコード

PP ロード問合せ情報から S7F1 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F1(ref DSHMSG smsg, TPPINQ_INFO info)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

info

PP ロード問合せ情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info の PP ロード問合せ情報から、引数 smsg に S7F1 メッセージを生成します。

2. 32. 2 DecodeS7F1() - S7F1 のデコード

DSHMSG 構造体に含まれる S7F1 メッセージをデコードして、PP ロード問合せ情報を info に格納します。

【構文】

```
public static int DecodeS7F1(ref DSHMSG smsg, ref TPPINQ_INFO info)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

info

PP ロード問合せ情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F1 メッセージをデコードして、PP ロード問合せ情報を info に格納します。

2. 32. 3 EncodeS7F2() - S7F2 のエンコード

与えられた ACK から S7F2 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F2(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S7F2 メッセージを生成します。

2. 32. 4 DecodeS7F2() - S7F2 のデコード

DSHMSG 構造体に含まれる S7F2 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS7F2( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 33 class_S7F3 - S7F3/S7F4 エンコード/デコードクラス

2. 33. 1 EncodeS7F3() - S7F3 のエンコード

PP 情報から S7F3 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F3(ref DSHMSG smsg, TPP_INFO info)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

info

PP 情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info の PP 情報から、引数 smsg に S7F3 メッセージを生成します。

2. 33. 2 DecodeS7F3() - S7F3 のデコード

DSHMSG 構造体に含まれる S7F3 メッセージをデコードして、PP 情報を info に格納します。

【構文】

```
public static int DecodeS7F3(ref DSHMSG smsg, ref TPP_INFO info)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

info

PP 情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F3 メッセージをデコードして、PP 情報を info に格納します。

2. 33. 3 EncodeS7F4() - S7F4 のエンコード

与えられた ACK から S7F4 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F4(ref DSHMSG smsg, int ack)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 smsg に S7F4 メッセージを生成します。

2. 33. 4 DecodeS7F4() - S7F4 のデコード

DSHMSG 構造体に含まれる S7F4 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS7F4( ref DSHMSG smsg, ref int ack)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 34 class_S7F5 - S7F5/S7F6 エンコード/デコードクラス

2. 34. 1 EncodeS7F5() - S7F5 のエンコード

PPID から S7F5 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F5( ref DSHMSG smsg, string ppid)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

ppid

PPID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 ppid から、引数 smsg に S7F5 メッセージを生成します。

2. 34. 2 DecodeS7F5() - S7F5 のデコード

DSHMSG 構造体に含まれる S7F5 メッセージをデコードして、PPID を取得します。

【構文】

```
public static int DecodeS7F5(ref DSHMSG smsg, ref string ppid)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

ppid

PPID を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F5 メッセージをデコードして、PPID を引数の ppid に格納します。

2. 34. 3 EncodeS7F6() - S7F6 のエンコード

与えられた ACK から S7F6 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F5(ref DSHMSG msg, string ppid)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ppid

PP 情報を取り出すための PPID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ppid の PP 情報を取り出し、その情報から引数 msg に S7F6 メッセージを生成します。

2. 34. 4 DecodeS7F6() - S7F6 のデコード

DSHMSG 構造体に含まれる S7F6 メッセージをデコードし、PP 情報を取得します。

【構文】

```
public static int DecodeS7F6(ref DSHMSG msg, TPP_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

PP 情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の info に PP 情報を格納します。

2. 35 class_S7F17 - S7F17/S7F18 エンコード/デコードクラス

2. 35. 1 EncodeS7F17() - S7F17のエンコード

PPID リストから S7F17 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F17(ref DSHMSG smsg, TPPID_LIST list)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

list

PPID が保存されているリストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list の PPID リストから、引数 smsg に S7F17 メッセージを生成します。

2. 35. 2 DecodeS7F17() - S7F17のデコード

DSHMSG 構造体に含まれる S7F17 メッセージをデコードして、PPID リストを list に格納します。

【構文】

```
public static int DecodeS7F17(ref DSHMSG smsg, ref TPP_INFO list)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

list

PPID を保存するリストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F17 メッセージをデコードして、PPID リストを引数の list に格納します。

2. 35. 3 EncodeS7F18() - S7F18 のエンコード

与えられた ACK から S7F18 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F18(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S7F18 メッセージを生成します。

2. 35. 4 DecodeS7F18() - S7F18 のデコード

DSHMSG 構造体に含まれる S7F18 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS7F18( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 36 class_S7F19 - S7F19/S7F20 エンコード/デコードクラス

2. 36. 1 EncodeS7F19() - S7F19 のエンコード

S7F19 メッセージを DSHMSG 構造体の中に生成します。(Header のみ)

【構文】

```
public static int EncodeS7F19( ref DSHMSG msg)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 msg に S7F19 メッセージを生成します。

2. 36. 2 EncodeS7F20() - S7F20 のエンコード

PPID リストから S7F20 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F20(ref DSHMSG msg, TPPID_LIST list)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

list

PPID リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 list から、引数 msg に S7F20 メッセージを生成します。

2. 36. 3 DecodeS7F20() - S7F20 のデコード

DSHMSG 構造体に含まれる S7F20 メッセージをデコードし、PPID リストに格納します。

【構文】

```
public static int DecodeS7F20(ref DSHMSG msg, ref TPPID_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

PPID を格納する配列リストです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の list の PPID リストに格納します。

2. 37 class_S7F23 - S7F23/S7F24 エンコード/デコードクラス

2. 37. 1 EncodeS7F23() - S7F23 のエンコード

FPP 情報から S7F23 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F23(ref DSHMSG msg, TFPP_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

FPP 情報が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 info の FPP 情報から、引数 msg に S7F23 メッセージを生成します。

2. 37. 2 DecodeS7F23() - S7F23 のデコード

DSHMSG 構造体に含まれる S7F23 メッセージをデコードして、FPP 情報を info に格納します。

【構文】

```
public static int DecodeS7F23(ref DSHMSG msg, ref TFPP_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

FPP 情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F23 メッセージをデコードして、FPP 情報を info に格納します。

2. 37. 3 EncodeS7F24() - S7F24 のエンコード

与えられた ACK から S7F24 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F24(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S7F24 メッセージを生成します。

2. 37. 4 DecodeS7F24() - S7F24 のデコード

DSHMSG 構造体に含まれる S7F24 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS7F24( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 38 class_S7F25 - S7F25/S7F26 エンコード/デコードクラス

2. 38. 1 EncodeS7F25() - S7F25 のエンコード

FPPID から S7F25 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F25( ref DSHMSG smsg, string ppid)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

ppid

FPPID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 ppid から、引数 smsg に S7F25 メッセージを生成します。

2. 38. 2 DecodeS7F25() - S7F25 のデコード

DSHMSG 構造体に含まれる S7F25 メッセージをデコードして、FPPID を取得します。

【構文】

```
public static int DecodeS7F25(ref DSHMSG smsg, ref string ppid)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

ppid

FPPID を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F25 メッセージをデコードして、FPPID を引数の ppid に格納します。

2. 38. 3 EncodeS7F26() - S7F26 のエンコード

与えられた ACK から S7F26 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F25(ref DSHMSG smsg, string ppid)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ppid

FPP 情報を取り出すための FPPID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ppid の FPP 情報を取り出し、その情報から引数 smsg に S7F26 メッセージを生成します。

2. 38. 4 DecodeS7F26() - S7F26 のデコード

DSHMSG 構造体に含まれる S7F26 メッセージをデコードし、FPP 情報を取得します。

【構文】

```
public static int EncodeS7F26(ref DSHMSG smsg, TFPP_INFO info)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

info

FPP 情報を格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして引数の info に FPP 情報を格納します。

2. 39 class_S7F27 - S7F27/S7F28 エンコード/デコードクラス

2. 39. 1 EncodeS7F27() - S7F27 のエンコード

PP 妥当性情報リストから S7F27 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F27(ref DSHMSG smsg, TPP_PVS_LIST list)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

list

PP 妥当性情報リストが保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(list=null)

【説明】

引数 list の PP 情報から、引数 smsg に S7F27 メッセージを生成します。

2. 39. 2 DecodeS7F27() - S7F27 のデコード

DSHMSG 構造体に含まれる S7F27 メッセージをデコードして、PP 妥当性情報リストを list に格納します。

【構文】

```
public static int DecodeS7F27(ref DSHMSG smsg, ref TPP_PVS_LIST list)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

list

PP 妥当性情報リストを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F27 メッセージをデコードして、PP 妥当性情報リストを list に格納します。

2. 39. 3 EncodeS7F28() - S7F28 のエンコード

与えられた ACK から S7F28 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F28(ref DSHMSG smsg, int ack)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 smsg に S7F28 メッセージを生成します。

2. 39. 4 DecodeS7F28() - S7F28 のデコード

DSHMSG 構造体に含まれる S7F28 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS7F28( ref DSHMSG smsg, ref int ack)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 smsg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 40 class_S7F29 - S7F29/S7F30 エンコード/デコードクラス

2. 40. 1 EncodeS7F29() - S7F29 のエンコード

PP 妥当性情報 (LENGTH) から S7F29 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F29(ref DSHMSG msg, int length)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

length

PP 妥当性問合せ情報 LENGTH が保存されています。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 length から、引数 msg に S7F29 メッセージを生成します。

2. 40. 2 DecodeS7F29() - S7F29 のデコード

DSHMSG 構造体に含まれる S7F29 メッセージをデコードして、PP 妥当性情報 (LENGTH) を length に格納します。

【構文】

```
public static int DecodeS7F29(ref DSHMSG msg, ref int length)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

length

PP 妥当性情報 (LENGTH) を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S7F29 メッセージをデコードして、PP 妥当性情報 (LENGTH) を length に格納します。

2. 40. 3 EncodeS7F30() - S7F30 のエンコード

与えられた ACK から S7F30 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS7F30(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S7F30 メッセージを生成します。

2. 40. 4 DecodeS7F30() - S7F30 のデコード

DSHMSG 構造体に含まれる S7F30 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS7F30( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 41 class_S10F1 - S10F1/S10F2 エンコード/デコードクラス

2. 41. 1 EncodeS10F1() - S10F1 のエンコード

端末 ID とテキストメッセージから S10F1 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS10F1( ref DSHMSG msg, int tid, string text)
```

【引数】

msg
エンコード先のメッセージ保存構造体です。

tid
端末 ID です。

text
テキストメッセージです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(length=null)

【説明】

引数 tid, text から、引数 msg に S10F1 メッセージを生成します。

2. 41. 2 DecodeS10F1() - S10F1 のデコード

DSHMSG 構造体に含まれる S10F1 メッセージをデコードして、端末メッセージ情報を取得します。

【構文】

```
public static int DecodeS10F1(ref DSHMSG msg, ref TTERM_INFO info)
```

【引数】

msg
デコード元のメッセージ保存構造体です。

info
端末 ID とテキストメッセージを保存するため TTERM_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S10F1 メッセージをデコードして、引数 info に端末 ID とメッセージを格納します。

2. 41. 3 EncodeS10F2() - S10F2 のエンコード

与えられた ACK から S10F2 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS10F2(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S10F2 メッセージを生成します。

2. 41. 4 DecodeS10F2() - S10F2 のデコード

DSHMSG 構造体に含まれる S10F2 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS10F2( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 42 class_S10F3 - S10F3/S10F4 エンコード/デコードクラス

2. 42. 1 EncodeS10F3() - S10F3 のエンコード

端末 ID とテキストメッセージから S10F3 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS10F3( ref DSHMSG msg, int tid, string text)
```

【引数】

msg エンコード先のメッセージ保存構造体です。
tid 端末 ID です。
text テキストメッセージです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(length=null)

【説明】

引数 tid, text から、引数 msg に S10F3 メッセージを生成します。

2. 42. 2 DecodeS10F3() - S10F3 のデコード

DSHMSG 構造体に含まれる S10F3 メッセージをデコードして、端末メッセージ情報を取得します。

【構文】

```
public static int DecodeS10F3(ref DSHMSG msg, ref TTERM_INFO info)
```

【引数】

msg デコード元のメッセージ保存構造体です。
info 端末 ID とテキストメッセージを保存するため TTERM_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S10F3 メッセージをデコードして、引数 info に端末 ID とメッセージを格納します。

2. 42. 3 EncodeS10F4() - S10F4 のエンコード

与えられた ACK から S10F4 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS10F4(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S10F4 メッセージを生成します。

2. 42. 4 DecodeS10F4() - S10F4 のデコード

DSHMSG 構造体に含まれる S10F4 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS10F4( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 43 class_S10F5 - S10F5/S10F6 エンコード/デコードクラス

2. 43. 1 EncodeS10F5() - S10F5 のエンコード

端末 ID とテキストメッセージから S10F5 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS10F5( ref DSHMSG smsg, TTERM_MULTI_INFO info)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

info

端末 ID と複数のメッセージのリストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(length=null)

【説明】

引数 info から、引数 smsg に S10F5 メッセージを生成します。

2. 43. 2 DecodeS10F5() - S10F5 のデコード

DSHMSG 構造体に含まれる S10F5 メッセージをデコードして、端末メッセージ情報を取得します。

【構文】

```
public static int DecodeS10F5(ref DSHMSG smsg, ref TTERM_MULTI_INFO info)
```

【引数】

smsg

デコード元のメッセージ保存構造体です。

info

端末 ID と複数のテキストメッセージを保存するため TTERM_MULTI_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S10F5 メッセージをデコードして、引数 info に端末 ID と複数のテキストメッセージを格納します。

2. 43. 3 EncodeS10F6() - S10F6 のエンコード

与えられた ACK から S10F6 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS10F6(ref DSHMSG msg, int ack)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

ack

応答 ACK です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 ack から、引数 msg に S10F6 メッセージを生成します。

2. 43. 4 DecodeS10F6() - S10F6 のデコード

DSHMSG 構造体に含まれる S10F6 メッセージをデコードし、応答コードを取得します。

【構文】

```
public static int DecodeS10F6( ref DSHMSG msg, ref int ack)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

ack

応答コードを格納するメモリです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の ack に応答コードを格納します。

2. 44 class_S14F9 - S14F9/S14F10 エンコード/デコードクラス

2. 44. 1 EncodeS14F9() - S14F9 のエンコード

TCJ_INFO のインスタンス情報 info から S14F9 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS14F9( ref DSHMSG msg, TCJ_INFO info )
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TCJ_INFO クラス(CJ 情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 info から引数 msg に S14F9 メッセージを生成します。

2. 44. 2 DecodeS14F9() - S14F9 のデコード

DSHMSG 構造体に含まれる S14F9 メッセージをデコードして CJ 情報を取得します。

【構文】

```
public static int DecodeS14F9(ref DSHMSG msg, ref TCJ_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

デコード結果を保存する TCJ_INFO クラス(CJ 情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S14F9 メッセージをデコードして、CJ 情報を引数 info に取得します。

2. 44. 3 EncodeS14F10() - S14F10のエンコード

与えられた応答情報から S14F10 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS14F10( ref DSHMSG msg, TOBJ_S14_ERR_INFO erS14info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

erS14info

CJ 生成に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 erS14info から、引数 msg に S14F10 メッセージを生成します。

2. 44. 4 DecodeS14F10() - S14F10のデコード

DSHMSG 構造体に含まれる S14F10 メッセージをデコードし、TOBJ_S14_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS14F10( ref DSHMSG msg, ref TOBJ_S14_ERR_INFO erS14info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

erS14info

CJ 生成に対する応答情報を保存するための TOBJ_S14_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして CJ 情報を引数 erS14info に保存します。

2. 45 class_S14F11 - S14F11/S14F12 エンコード/デコードクラス

2. 45. 1 EncodeS14F11() - S14F11 のエンコード

TCJ_INFO のインスタンス情報 info から S14F11 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS14F11( ref DSHMSG msg, TCJ_INFO info )
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TCJ_INFO クラス(CJ 情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 info から引数 msg に S14F11 メッセージを生成します。

2. 45. 2 DecodeS14F11() - S14F11 のデコード

DSHMSG 構造体に含まれる S14F11 メッセージをデコードして CJ 情報を取得します。

【構文】

```
public static int DecodeS14F11(ref DSHMSG msg, ref TCJ_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

デコード結果を保存する TCJ_INFO クラス(CJ 情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S14F11 メッセージをデコードして、CJ 情報を引数 info に取得します。

2. 45. 3 EncodeS14F12() - S14F12 のエンコード

与えられた応答情報から S14F12 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS14F12( ref DSHMSG msg, TOBJ_S14_ERR_INFO erS14info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

erS14info

CJ 削除に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 erS14info から、引数 msg に S14F12 メッセージを生成します。

2. 45. 4 DecodeS14F12() - S14F12 のデコード

DSHMSG 構造体に含まれる S14F12 メッセージをデコードし、TOBJ_S14_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS14F12( ref DSHMSG msg, ref TOBJ_S14_ERR_INFO erS14info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

erS14info

CJ 削除に対する応答情報を保存するための TOBJ_S14_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして CJ 情報を引数 erS14info に保存します。

2. 46 class_S15F3 - S15F3/S15F4 エンコード/デコードクラス

2. 46. 1 EncodeS15F3() - S15F3 のエンコード

TRCP_ACT_INFO のインスタンス情報 info から S15F3 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F3( ref DSHMSG msg, TRCP_ACT_INFO info )
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TRCP_ACT_INFO クラス(レシピアクション情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 info から引数 msg に S15F3 メッセージを生成します。

2. 46. 2 DecodeS15F3() - S15F3 のデコード

DSHMSG 構造体に含まれる S15F3 メッセージをデコードしてレシピアクション情報を取得します。

【構文】

```
public static int DecodeS15F3(ref DSHMSG msg, ref TRCP_ACT_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

デコード結果を保存する TRCP_ACT_INFO クラス(レシピアクション情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S15F3 メッセージをデコードして、レシピアクション情報を引数 info に取得します。

2. 46. 3 EncodeS15F4() - S15F4 のエンコード

与えられた応答情報から S15F4 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F4( ref DSHMSG msg, TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

レスポンスアクションに対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S15F4 メッセージを生成します。

2. 46. 4 DecodeS15F4() - S15F4 のデコード

DSHMSG 構造体に含まれる S15F4 メッセージをデコードし、TCJ_CMD_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS15F4( ref DSHMSG msg, ref TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

レスポンスアクションに対する応答情報を保存するための TCJ_CMD_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてレスポンスアクション情報を引数 rsp_info に保存します。

2. 47 class_S15F5 - S15F5/S15F6 エンコード/デコードクラス

2. 47. 1 EncodeS15F5() - S15F5 のエンコード

TRCP_RENAME_INFO のインスタンス情報 info から S15F5 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F5( ref DSHMSG msg, TRCP_RENAME_INFO info )
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TRCP_RENAME_INFO クラス(レシピリネームアクション情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(id=null)

【説明】

引数 info から引数 msg に S15F5 メッセージを生成します。

2. 47. 2 DecodeS15F5() - S15F5 のデコード

DSHMSG 構造体に含まれる S15F5 メッセージをデコードしてレシピリネームアクション情報を取得します。

【構文】

```
public static int DecodeS15F5(ref DSHMSG msg, ref TRCP_RENAME_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

デコード結果を保存する TRCP_RENAME_INFO クラス(レシピリネームアクション情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S15F5 メッセージをデコードして、レシピリネームアクション情報を引数 info に取得します。

2. 47. 3 EncodeS15F6() - S15F6 のエンコード

与えられた応答情報から S15F6 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F6( ref DSHMSG msg, TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

レスポンスアクションに対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S15F6 メッセージを生成します。

2. 47. 4 DecodeS15F6() - S15F6 のデコード

DSHMSG 構造体に含まれる S15F6 メッセージをデコードし、TCJ_CMD_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS15F6( ref DSHMSG msg, ref TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

レスポンスアクションに対する応答情報を保存するための TCJ_CMD_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてレスポンスアクション情報を引数 rsp_info に保存します。

2. 48 class_S15F7 - S15F7/S15F8 エンコード/デコードクラス

2. 48. 1 EncodeS15F7() - S15F7 のエンコード

レシピ ID から S15F7 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F7( ref DSHMSG msg, string objspec)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

objspec

レシピ ID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 objspec から、引数 msg に S15F7 メッセージを生成します。

2. 48. 2 DecodeS15F7() - S15F7 のデコード

DSHMSG 構造体に含まれる S15F7 メッセージをデコードして、レシピ ID を取得します。

【構文】

```
public static int DecodeS15F7(ref DSHMSG msg, ref string objspec)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

objspec

レシピ ID を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S15F7 メッセージをデコードして、レシピ ID を引数の objspec に格納します。

2. 48. 3 EncodeS15F8() - S15F8 のエンコード

与えられたレシピスペースデータから S15F8 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F8(ref DSHMSG msg, TRCP_S15F8_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

レシピスペースデータが保存されている TRCP_S15F8_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info のレシピスペースデータ情報から引数 msg に S15F8 メッセージを生成します。

2. 48. 4 DecodeS15F8() - S15F8 のデコード

DSHMSG 構造体に含まれる S15F8 メッセージをデコードし、レシピスペースデータ情報を取得します。

【構文】

```
public static int DecodeS15F8(ref DSHMSG msg, ref TRCP_S15F8_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

レシピスペースデータ情報を保存する TRCP_S15F8_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の info にレシピスペースデータ情報を格納します。

2. 49 class_S15F9 - S15F9/S15F10 エンコード/デコードクラス

2. 49. 1 EncodeS15F9() - S15F9 のエンコード

レシピ ID から S15F9 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F9( ref DSHMSG msg, string objspec)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

objspec

レシピ ID です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。(info=null)

【説明】

引数 objspec から、引数 msg に S15F9 メッセージを生成します。

2. 49. 2 DecodeS15F9() - S15F9 のデコード

DSHMSG 構造体に含まれる S15F9 メッセージをデコードして、レシピ ID を取得します。

【構文】

```
public static int DecodeS15F9(ref DSHMSG msg, ref string objspec)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

objspec

レシピ ID を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

DSHMSG 構造体に含まれる S15F9 メッセージをデコードして、レシピ ID を引数の objspec に格納します。

2. 49. 3 EncodeS15F10() - S15F10 のエンコード

与えられたレシピステータスデータから S15F10 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F10(ref DSHMSG msg, TRCP_S15F10_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

レシピステータスデータが保存されている TRCP_S15F10_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info のレシピステータスデータ情報から引数 msg に S15F10 メッセージを生成します。

2. 49. 4 DecodeS15F10() - S15F10 のデコード

DSHMSG 構造体に含まれる S15F10 メッセージをデコードし、レシピステータスデータ情報を取得します。

【構文】

```
public static int DecodeS15F10( ref DSHMSG msg, ref TRCP_S15F10_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

レシピステータスデータ情報を保存する TRCP_S15F10 クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードして引数の info にレシピステータスデータ情報を格納します。

2. 50 class_S15F13 - S15F13/S15F14 エンコード/デコードクラス

2. 50. 1 EncodeS15F13() - S15F13のエンコード

TRCP_INFO のインスタンス情報 info から S15F13 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F13(ref DSHMSG msg, TRCP_INFO info, bool updt_flag)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TRCP_INFO クラス(レシピ情報)のインスタンスです。

updt_flag

生成/更新(false/true)を指定します。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 info と updt_flag から引数 msg に S15F13 メッセージを生成します。

2. 50. 2 DecodeS15F13() - S15F13のデコード

DSHMSG 構造体に含まれる S15F13 メッセージをデコードしてレシピ情報を取得します。

【構文】

```
public static int DecodeS15F13(ref DSHMSG msg, ref TRCP_INFO info, ref bool updt_flag)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

デコード結果を保存する TRCP_INFO クラス(レシピ情報)のインスタンスです。

updt_flag

生成/更新(false/true)を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S15F13 メッセージをデコードして、レシピ情報を引数 info と updt_flag に取得します。

2. 50. 3 EncodeS15F14() - S15F14 のエンコード

与えられた応答情報から S15F14 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F14( ref DSHMSG msg, TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

レシピ生成/更新に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S15F14 メッセージを生成します。

2. 50. 4 DecodeS15F14() - S15F14 のデコード

DSHMSG 構造体に含まれる S15F14 メッセージをデコードし、TCJ_CMD_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS15F14( ref DSHMSG msg, ref TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

レシピ生成/更新に対する応答情報を保存するための TCJ_CMD_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてレシピ情報を引数 rsp_info に保存します。

2. 51 class_S15F17 - S15F17/S15F18 エンコード/デコードクラス

2. 51. 1 EncodeS15F17() - S15F17 のエンコード

TRCP_RETRIEVE_INFO のインスタンス情報 info から S15F17 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F17(ref DSHMSG msg, TRCP_RETRIEVE_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TRCP_RETRIEVE_INFO クラス(レシピ情報)のインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 info から引数 msg に S15F17 メッセージを生成します。

2. 51. 2 DecodeS15F17() - S15F17 のデコード

DSHMSG 構造体に含まれる S15F17 メッセージをデコードしてレシピ情報を取得します。

【構文】

```
public static int DecodeS15F17(ref DSHMSG msg, ref string rcpid, ref int sect)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rcpid

レシピ ID を保存します。

sect

レシピセクションを保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S15F17 メッセージをデコードして、レシピ ID とセクションを引数 rcpid と sect に取得します。

2. 51. 3 EncodeS15F18() - S15F18 のエンコード

与えられた応答情報から S15F18 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS15F18( ref DSHMSG msg, TRCP_S15F18_INFO info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

info

レシピ情報に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 info から、引数 msg に S15F18 メッセージを生成します。

2. 51. 4 DecodeS15F18() - S15F18 のデコード

DSHMSG 構造体に含まれる S15F18 メッセージをデコードし、TRCP_S15F18_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS15F18( ref DSHMSG msg, ref TRCP_S15F18_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

レシピ ID、セクション応答情報を保存するための TRCP_S15F18_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてレシピ情報を引数 info に保存します。

2. 52 class_S16F5 - S16F5/S16F6 エンコード/デコードクラス

2. 52. 1 EncodeS16F5() - S16F5 のエンコード

TPRJ_CMD_INFO クラスのインスタンス情報 info から S16F5 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F5(ref DSHMSG msg, TPRJ_CMD_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TPRJ_CMD_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 info から引数 msg に S16F5 メッセージを生成します。

2. 52. 2 DecodeS16F5() - S16F5 のデコード

DSHMSG 構造体に含まれる S16F5 メッセージをデコードしてプロセスジョブコマンド情報を取得します。

【構文】

```
public static int DecodeS16F5(ref DSHMSG msg, ref TPRJ_CMD_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

プロセスジョブコマンド情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S16F5 メッセージをデコードして、プロセスジョブコマンド情報を引数 info に取得します。

2. 52. 3 EncodeS16F6() - S16F6 のエンコード

与えられた応答情報から S16F6 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F6( ref DSHMSG msg, TPRJ_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

プロセスジョブコマンドに対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S16F6 メッセージを生成します。

2. 52. 4 DecodeS16F6() - S16F6 のデコード

DSHMSG 構造体に含まれる S16F6 メッセージをデコードし、TPRJ_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS16F6( ref DSHMSG msg, ref TPRJ_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

プロセスジョブコマンドに対する応答情報を保存するための TPRJ_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてプロセスジョブコマンドに対する応答情報を引数 rsp_info に保存します。

2. 53 class_S16F11 - S16F11/S16F12 エンコード/デコードクラス

2. 53. 1 EncodeS16F11() - S16F11 のエンコード

TPRJ_INFO クラスのインスタンス情報 info から S16F11 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F11(ref DSHMSG msg, TPRJ_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TPRJ_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 info から引数 msg に S16F11 メッセージを生成します。

2. 53. 2 DecodeS16F11() - S16F11 のデコード

DSHMSG 構造体に含まれる S16F11 メッセージをデコードしてプロセスジョブ情報を取得します。

【構文】

```
public static int DecodeS16F11(ref DSHMSG msg, ref TPRJ_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

プロセスジョブ情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S16F11 メッセージをデコードして、プロセスジョブ情報を引数 info に取得します。

2. 53. 3 EncodeS16F12() - S16F12のエンコード

与えられた応答情報から S16F12 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F12( ref DSHMSG msg, TPRJ_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

プロセスジョブ生成に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S16F12 メッセージを生成します。

2. 53. 4 DecodeS16F12() - S16F12のデコード

DSHMSG 構造体に含まれる S16F12 メッセージをデコードし、TPRJ_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS16F12( ref DSHMSG msg, ref TPRJ_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

プロセスジョブ情報生成に対する応答情報を保存するための TPRJ_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてプロセスジョブ生成情報に対する応答情報を引数 rsp_info に保存します。

2. 54 class_S16F15 - S16F15/S16F16 エンコード/デコードクラス

2. 54. 1 EncodeS16F15() - S16F15 のエンコード

TPRJ_LIST クラスのインスタンス情報 list から S16F15 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F15(ref DSHMSG msg, TPRJ_LIST list)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

list

TPRJ_LIST クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 list から引数 msg に S16F15 メッセージを生成します。

2. 54. 2 DecodeS16F15() - S16F15 のデコード

DSHMSG 構造体に含まれる S16F15 メッセージをデコードしてマルチプロセスジョブ情報を取得します。

【構文】

```
public static int DecodeS16F15(ref DSHMSG msg, ref TPRJ_LIST list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

list

マルチプロセスジョブ情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S16F15 メッセージをデコードして、マルチプロセスジョブ情報を引数 list に取得します。

2. 54. 3 EncodeS16F16() - S16F16 のエンコード

与えられた応答情報から S16F16 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F16( ref DSHMSG msg, TMPRJ_ERR_INFO rsp_list)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_list

マルチプロセスジョブ生成に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_list から、引数 msg に S16F16 メッセージを生成します。

2. 54. 4 DecodeS16F16() - S16F16 のデコード

DSHMSG 構造体に含まれる S16F16 メッセージをデコードし、TMPRJ_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS16F16( ref DSHMSG msg, ref TMPRJ_ERR_INFO rsp_list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_list

マルチプロセスジョブ情報生成に対する応答情報を保存するための TMPRJ_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてマルチプロセスジョブ生成情報に対する応答情報を引数 rsp_list に保存します。

2. 55 class_S16F17 - S16F17/S16F18 エンコード/デコードクラス

2. 55. 1 EncodeS16F17() - S16F17のエンコード

TPRJ_DEQ_INFO クラスのインスタンス情報 info から S16F17 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F17(ref DSHMSG msg, TPRJ_DEQ_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TPRJ_DEQ_INFO クラスのインスタンスです。(複数のプロセスジョブの削除リスト)

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 info から引数 msg に S16F17 メッセージを生成します。

2. 55. 2 DecodeS16F17() - S16F17のデコード

DSHMSG 構造体に含まれる S16F17 メッセージをデコードしてプロセスジョブ削除情報を取得します。

【構文】

```
public static int DecodeS16F17(ref DSHMSG msg, ref TPRJ_DEQ_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

プロセスジョブ削除情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S16F17 メッセージをデコードして、プロセスジョブ削除情報を引数 info に取得します。

2. 55. 3 EncodeS16F18() - S16F18 のエンコード

与えられた応答情報から S16F18 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F18( ref DSHMSG msg, TMPRJ_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

プロセスジョブ削除に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S16F18 メッセージを生成します。

2. 55. 4 DecodeS16F18() - S16F18 のデコード

DSHMSG 構造体に含まれる S16F18 メッセージをデコードし、TMPRJ_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS16F18( ref DSHMSG msg, ref TMPRJ_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

プロセスジョブ削除に対する応答情報を保存するための TMPRJ_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてプロセスジョブ削除に対する応答情報を引数 rsp_info に保存します。

2. 56 class_S16F19 - S16F19/S16F20 エンコード/デコードクラス

2. 56. 1 EncodeS16F19() - S16F19 のエンコード

S16F19 メッセージを DSHMSG 構造体の中に生成します。(Header のみで Text なし)

【構文】

```
public static int EncodeS16F19(ref DSHMSG smsg)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 smsg に S16F19 メッセージを生成します。

2. 56. 2 EncodeS16F20() - S16F20 のエンコード

与えられた応答情報から S16F20 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F20( ref DSHMSG smsg, TPRJ_STATE_LIST rsp_list)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_list

登録されているプロセスジョブの状態情報リストです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_list から、引数 smsg に S16F20 メッセージを生成します。

2. 56. 3 DecodeS16F20() - S16F20 のデコード

DSHMSG 構造体に含まれる S16F20 メッセージをデコードし、TPRJ_STATE_LIST のインスタンスに取得します。

【構文】

```
public static int DecodeS16F20( ref DSHMSG msg, ref TPRJ_STATE_LIST rsp_list)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_list

登録されているプロセスジョブの状態情報を保存するための TPRJ_STATE_LIST のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてプロセスジョブの状態情報問い合わせに対する応答情報を引数 rsp_list に保存します。

2. 57 class_S16F21 - S16F21/S16F22 エンコード/デコードクラス

2. 57. 1 EncodeS16F21() - S16F21 のエンコード

S16F21 メッセージを DSHMSG 構造体の中に生成します。(Header のみで Text なし)

【構文】

```
public static int EncodeS16F21(ref DSHMSG smsg)
```

【引数】

smsg

エンコード先のメッセージ保存構造体です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 smsg に S16F21 メッセージを生成します。

2. 57. 2 EncodeS16F22() - S16F22 のエンコード

与えられた応答情報から S16F22 メッセージを DSHMSG 構造体の中に生成します。
登録可能のプロセスジョブの数をお応答します。

【構文】

```
public static int EncodeS16F22( ref DSHMSG smsg)
```

【引数】

smsg

エンコードしたメッセージを保存する DSHMSG 構造体です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

DSHEng5 は、プロセスジョブ登録領域の空き領域数をもとめて、そのスペース数から、引数 smsg に S16F22 メッセージを生成します。

2. 57. 3 DecodeS16F22() - S16F22 のデコード

DSHMSG 構造体に含まれる S16F22 メッセージをデコードし、プロセスジョブ保存領域の空き数を取得します。

【構文】

```
public static int DecodeS16F22( ref DSHMSG msg, ref int prj_space)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

prj_space

プロセスジョブ保存領域の空き数を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてプロセスジョブの保存領域の空き数を引数 prj_space に取得します。

2. 58 class_S16F27 - S16F27/S16F28 エンコード/デコードクラス

2. 58. 1 EncodeS16F27() - S16F27 のエンコード

TCJ_CMD_INFO クラスのインスタンス情報 info から S16F27 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F27(ref DSHMSG msg, TCJ_CMD_INFO info)
```

【引数】

msg

エンコード先のメッセージ保存構造体です。

info

TCJ_CMD_INFO クラスのインスタンスです。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 info から引数 msg に S16F27 メッセージを生成します。

2. 58. 2 DecodeS16F27() - S16F27 のデコード

DSHMSG 構造体に含まれる S16F27 メッセージをデコードして CJ コマンド関連情報を取得します。

【構文】

```
public static int DecodeS16F27(ref DSHMSG msg, ref TCJ_CMD_INFO info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

info

レシピ関連情報を保存します。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg 含まれる S16F27 メッセージをデコードして、CJ コマンド関連情報を引数 info に取得します。

2. 58. 3 EncodeS16F28() - S16F28 のエンコード

与えられた応答情報から S16F28 メッセージを DSHMSG 構造体の中に生成します。

【構文】

```
public static int EncodeS16F28( ref DSHMSG msg, TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

エンコードしたメッセージを保存する DSHMSG 構造体です。

rsp_info

CJ コマンド関連に対する応答情報です。

【戻り値】

返却値	意 味
= 0	Encode できた。
(-1)	Encode できなかった。

【説明】

引数 rsp_info から、引数 msg に S16F28 メッセージを生成します。

2. 58. 4 DecodeS16F28() - S16F28 のデコード

DSHMSG 構造体に含まれる S16F28 メッセージをデコードし、TCJ_CMD_ERR_INFO のインスタンスに取得します。

【構文】

```
public static int DecodeS16F28( ref DSHMSG msg, ref TCJ_CMD_ERR_INFO rsp_info)
```

【引数】

msg

デコード元のメッセージ保存構造体です。

rsp_info

CJ コマンド関連に対する応答情報を保存するための TCJ_CMD_ERR_INFO のインスタンスです。

【戻り値】

返却値	意 味
= 0	Decode できた。
(-1)	Decode できなかった。

【説明】

引数 msg のメッセージをデコードしてレスポンスに対する応答情報を引数 rsp_info に保存します。