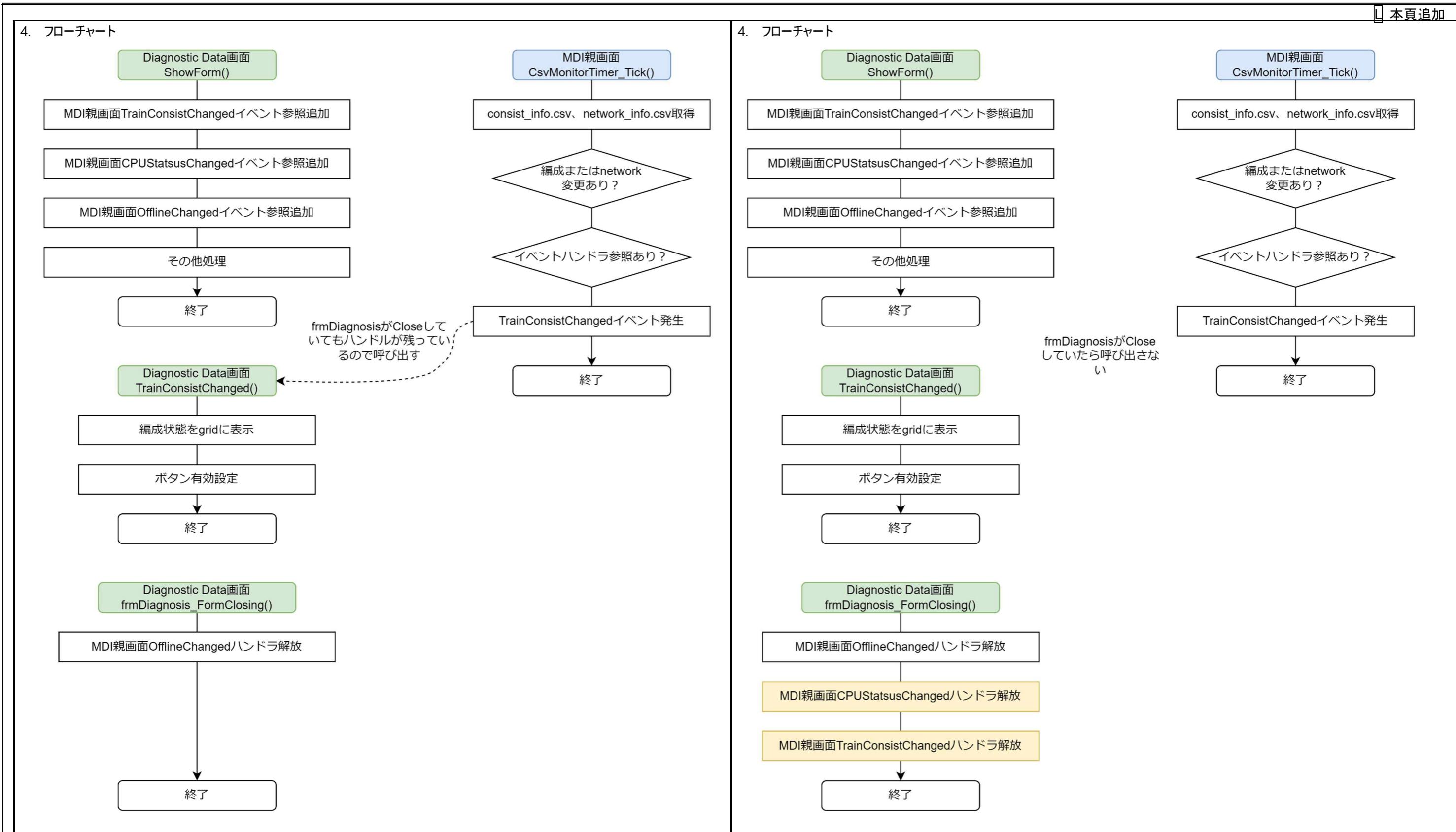


No.143 Pseudo Fault で Offline になったときの強制終了

| 変更点管理番号 No. | 143 |
|--|---|
| ソース・ファイル名 | frmCSTList.cs, frmDateSetting.cs, frmDiagnosis.cs, frmErasure.cs, frmFaultLoggingSuspension.cs, frmLogDataDownload.cs, frmMonitorIOStatus.cs, frmParameter.cs, frmPseudoFailure.cs, frmReboot.cs, frmSimpleCommand.cs, frmUploadUnitSelectionFMP.cs, frmUploadUnitSelectionSW.cs, frmVersionInformationBase.cs, TrainImageFlexGrid.cs |
| モジュール名 | MDS_R211PTE.exe |
| 修 正 内 容 | |
| Exception メッセージが表示されないようにする。また、Close された Form のイベントが呼び出されないようにイベントハンドラを解放しておく。 | |
| 修 正 前 | 修 正 後 |
| <p>1. Car 表示処理 SetTrainImage()で予期せぬエラーが発生して強制終了する。</p> <p>2. Offline になったときのイベント発生経路</p> <ol style="list-style-type: none"> 1) Diagnostic Data 画面を開くと MDI 親画面の TrainConsistChanged イベントに参照を追加する。 2) 編成が変化すると MDI 親画面が Diagnostic Data 画面の TrainConsistChanged イベントを発生させる。 3) Pseudo Fault 画面などへ遷移して Diagnostic Data 画面を Close()する。このとき MDI 親画面への参照を解除していない。この場合、.net framework はガベージコレクトせずメモリに残り続ける。 4) 車上とネットワーク切断して Offline になったとき、MDI 親画面が Diagnostic Data 画面の TrainConsistChanged イベントを発生させる。Diagnostic Data 画面が Close されてもメモリ上にいるため呼び出せる。 → Diagnostic Data 画面の TrainConsistChanged イベントで画面要素にアクセスしたときに null 参照が発生する。 <p>3. イベント参照画面</p> <p>CSTList Date Time Setting Diagnosis Erasure Fault Logging Suspension Data Monitoring Parameter Pseudo Fault Reboot Force Fitting Statistics SW Upload</p> | <p>1. Car 表示処理 SetTrainImage()で予期せぬエラーが発生して強制終了する。 エラーメッセージを出さないように try-catch でエラートラップする。</p> <p>2. Offline になったときのイベント発生経路</p> <ol style="list-style-type: none"> 1) Diagnostic Data 画面を開くと MDI 親画面の TrainConsistChanged イベントに参照を追加する。 2) 編成が変化すると MDI 親画面が Diagnostic Data 画面の TrainConsistChanged イベントを発生させる。 3) Pseudo Fault 画面などへ遷移して Diagnostic Data 画面を Close()する。このとき MDI 親画面への参照を解除する。解除すると.net framework がガベージコレクトしてメモリ上には残らない。 4) 車上とのネットワークを切断して Offline になったとき、MDI 親画面が Diagnostic Data 画面の TrainConsistChanged イベントを発生させる。ただし、Diagnostic Data 画面が Close されている場合は呼び出せない。 → Diagnostic Data 画面の TrainConsistChanged イベントは発生しないので null 参照も発生しない。 <p>3. イベント参照画面</p> <p>CSTList Date Time Setting Diagnosis Erasure Fault Logging Suspension Data Monitoring Parameter Pseudo Fault Reboot Force Fitting Statistics SW Upload</p> |

| | | | | | | | | | | |
|--|--|--|---------------------------------|--|--|--|--|--|----------------------|-------|
| | | | MITSUBISHI ELECTRIC CORPORATION | | | | | | | |
| | | | 改 定 CHANGE | | | | | | 文 書 番 号 DOCUMENT NO. | |
| | | | L | | | | | | JHCDF22 | 13-22 |



5. MDI 親ウインドウが通知するイベントハンドラ
イベントハンドラの追加と解放の処理を抜粋した。

1) TrainConsistChanged
frmDiagnosis.cs,_parent.TrainConsistChanged += _trainConsistChangedEvHandler;

frmLogDataDownload.cs,_parent.TrainConsistChanged += _trainConsistChangedEvHandler;
frmLogDataDownload.cs,_parent.TrainConsistChanged -= _trainConsistChangedEvHandler;

2) CPUStatusChanged
frmCTList.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmDateSettings.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmDiagnosis.cs,_parent.CPUStatusChanged += _cpuStsChangedHandler;

frmErasure.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmFaultLoggingSuspension.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmLogDataDownload.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmMonitorIOStatus.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmParameter.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmPseudoFailure.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmReboots.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmSimpleCommand.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmStatistics.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmUploadUnitSelectionFMP.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmUploadUnitSelectionSW.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

frmVersionInformationBase.cs,_parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);

5. MDI 親ウインドウが通知するイベントハンドラ
`protected override void OnFormClosed(FormClosedEventArgs e)` を実装してイベントハンドラを解放する。

1) TrainConsistChanged
frmDiagnosis.cs: _parent.TrainConsistChanged += _trainConsistChangedEvHandler;
frmDiagnosis.cs: _parent.TrainConsistChanged -= _trainConsistChangedEvHandler;
frmLogDataDownload.cs: _parent.TrainConsistChanged += _trainConsistChangedEvHandler;
frmLogDataDownload.cs: _parent.TrainConsistChanged -= _trainConsistChangedEvHandler;

2) CPUStatusChanged
frmCSTList.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmCSTList.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmDateSetting.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmDateSetting.cs, _parent.CPUStatusChanged -= _cpuStsChangedHandler;
frmDiagnosis.cs, _parent.CPUStatusChanged += _cpuStsChangedHandler;
frmDiagnosis.cs, _parent.CPUStatusChanged -= _cpuStsChangedHandler;
frmErasure.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmErasure.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmFaultLoggingSuspension.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmFaultLoggingSuspension.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmLogDataDownload.cs, _parent.CPUStatusChanged += _cpuStsChangedHandler;
frmLogDataDownload.cs, _parent.CPUStatusChanged -= _cpuStsChangedHandler;
frmMonitorIOStatus.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmMonitorIOStatus.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmParameter.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmParameter.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmPseudoFailure.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmPseudoFailure.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmReboots.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmReboots.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmSimpleCommands.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmSimpleCommands.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmStatistics.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmStatistics.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmUnitSelectionFMP.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmUnitSelectionFMP.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmUploadUnitSelectionSw.cs, _parent.CPUStatusChanged += this.CPUStatusChanged;
frmUploadUnitSelectionSw.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;
frmVersionInformationBase.cs, _parent.CPUStatusChanged += new EventHandler(this.CPUStatusChanged);
frmVersionInformationBase.cs, _parent.CPUStatusChanged -= this.CPUStatusChanged;

| | | | | | | | | | | |
|--|--|--|---------------------------------|--|--|--|--|--|--|--|
| | | | MITSUBISHI ELECTRIC CORPORATION | | | | | | | |
| | | | 改 定 CHANGE | | | | | | | |
| | | | L | | | | | | | |

3) OfflineChanged
frmCTLList.cs,_parent.OfflineChanged += new EventHandler(this.btnClose_Click);

frmDateSettings.cs,_parent.OfflineChanged += _offlineChangedHandler;
frmDateSettings.cs,_parent.OfflineChanged -= _offlineChangedHandler;
frmDiagnosis.cs,_parent.OfflineChanged += _offlineChangedHandler;
frmDiagnosis.cs,_parent.OfflineChanged -= _offlineChangedHandler;
frmErasure.cs,_parent.OfflineChanged += _offlineChangedHandler;
frmErasure.cs,_parent.OfflineChanged -= _offlineChangedHandler;
frmExecuteSelfTestMDCs.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmExecuteSelfTestMDCs.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmFaultLoggingSuspension.cs,_parent.OfflineChanged += _offlineChangedHandler;
frmFaultLoggingSuspension.cs,_parent.OfflineChanged -= _offlineChangedHandler;
frmLogDataDownload.cs,_parent.OfflineChanged += _returnClickEvHandler;
frmLogDataDownload.cs,_parent.OfflineChanged -= _returnClickEvHandler;
frmMonitorIOSTatus.cs,_parent.OfflineChanged += new EventHandler(this.btnClose_Click);

frmParameter.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmParameter.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmParameterExports.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmParameterExports.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmParameterImports.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmParameterImports.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmPseudoFailure.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmPseudoFailure.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmReboots.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmReboots.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmSimpleCommands.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmSimpleCommands.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmStatistics.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmStatistics.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmTestResults.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmTestResults.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmTestResultHelps.cs,_parent.OfflineChanged += _returnClickEvHandler;
frmTestResultHelps.cs,_parent.OfflineChanged -= _returnClickEvHandler;
frmUploadUnitSelectionFMP.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmUploadUnitSelectionFMP.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmUploadUnitSelectionSW.cs,_parent.OfflineChanged += _returnClickEvHandler;
frmUploadUnitSelectionSW.cs,_parent.OfflineChanged -= _returnClickEvHandler;
frmVersionInformationBase.cs,_parent.OfflineChanged += _closeClickEvHandler;
frmVersionInformationBase.cs,_parent.OfflineChanged -= _closeClickEvHandler;
frmVersionSelectionFMP.cs,_parent.OfflineChanged += _returnClickEvHandler;
frmVersionSelectionFMP.cs,_parent.OfflineChanged -= _returnClickEvHandler;
frmVersionSelectionSW.cs,_parent.OfflineChanged += _returnClickEvHandler;
frmVersionSelectionSW.cs,_parent.OfflineChanged -= _returnClickEvHandler;

3) OfflineChanged

```
frmCSTList.cs, _parent.OfflineChanged += new EventHandler(this.btnClose_Click);
frmCSTList.cs, _parent.OfflineChanged -= this.btnClose_Click;
frmDateSetting.cs, _parent.OfflineChanged += _offlineChangedHandler;
frmDateSetting.cs, _parent.OfflineChanged -= _offlineChangedHandler;
frmDiagnosis.cs, _parent.OfflineChanged += _offlineChangedHandler;
frmDiagnosis.cs, _parent.OfflineChanged -= _offlineChangedHandler;
frmErasure.cs, _parent.OfflineChanged += _offlineChangedHandler;
frmErasure.cs, _parent.OfflineChanged -= _offlineChangedHandler;
frmExecuteSelfTestMD.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmExecuteSelfTestMD.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmFaultLoggingSuspension.cs, _parent.OfflineChanged += _offlineChangedHandler;
frmFaultLoggingSuspension.cs, _parent.OfflineChanged -= _offlineChangedHandler;
frmLogDataDownload.cs, _parent.OfflineChanged += _returnClickEvHandler;
frmLogDataDownload.cs, _parent.OfflineChanged -= _returnClickEvHandler;
frmMonitorIOStatus.cs, _parent.OfflineChanged += new EventHandler(this.btnReturn_Click);
frmMonitorIOStatus.cs, _parent.OfflineChanged -= this.btnReturn_Click;
frmParameter.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmParameter.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmParameterExport.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmParameterExport.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmParameterImport.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmParameterImport.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmPseudoFailure.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmPseudoFailure.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmReboots.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmReboots.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmSimpleCommands.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmSimpleCommands.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmStatistics.cs, _parent.OfflineChanged += _returnClickEvHandler;
frmStatistics.cs, _parent.OfflineChanged -= _returnClickEvHandler;
frmTestResults.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmTestResults.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmTestResultHelp.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmTestResultHelp.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmUploadUnitSelectionFMP.cs, _parent.OfflineChanged += _returnClickEvHandler;
frmUploadUnitSelectionFMP.cs, _parent.OfflineChanged -= _returnClickEvHandler;
frmUploadUnitSelectionSW.cs, _parent.OfflineChanged += _returnClickEvHandler;
frmUploadUnitSelectionSW.cs, _parent.OfflineChanged -= _returnClickEvHandler;
frmVersionInformationBase.cs, _parent.OfflineChanged += _closeClickEvHandler;
frmVersionInformationBase.cs, _parent.OfflineChanged -= _closeClickEvHandler;
frmVersionSelectionFMP.cs, _parent.OfflineChanged += _returnClickEvHandler;
frmVersionSelectionFMP.cs, _parent.OfflineChanged -= _returnClickEvHandler;
frmVersionSelectionSW.cs, _parent.OfflineChanged += _returnClickEvHandler;
frmVersionSelectionSW.cs, _parent.OfflineChanged -= _returnClickEvHandler;
```

| | | | | | | | | | | |
|--|--|--|---------------------------------|--|--|--|--|--|--|--|
| | | | MITSUBISHI ELECTRIC CORPORATION | | | | | | | |
| | | | 改 定 CHANGE | | | | | | | |
| | | | L | | | | | | | |

No.145 USB Import の Snapshot 変換不良

| | | | |
|---|---|--|--|
| 変更点管理番号 No. | 145 | | |
| ソース・ファイル名 | clsUSBImport.cs | | |
| モジュール名 | MDS_R211PTE.exe | | |
| 修 正 内 容 | | | |
| MDS Event Log の USB Import において MDS Snapshot 変換のパス設定を修正 | | | |
| 修 正 前 | | | |
| <p>1. MDS Event Log の USB Import 処理</p> <ol style="list-style-type: none"> 1) USB Import ディレクトリ選択 2) データがある対象フォルダを検索「MMDDyyyyHHmmss_Car¥EVENTLOG」 3) 対象フォルダにある consist_info_EventLog.csv から編成情報取得 4) EVENTLOG フォルダをテンポラリにコピー 5) SNAPSHOT フォルダをテンポラリへコピーしない 6) MDS Snapshot 変換処理呼び出し ConvertTcmsSnapshot()の引数に Snapshot と EventLog の格納フォルダをセットする。 Snapshot フォルダ: {USB 対象フォルダ}¥03062025110721_4095¥SNAPSHOT EventLog フォルダ: {テンポラリ}¥4095¥MDSEventLog 7) Snapshot 変換(Snapshot ファイルがある同じ場所に.snap ファイルを作成) *USB に変換ファイルを作成してしまう 8) Snapshot のディレクトリ構造を維持したまま EVENTLOG と並列にある SNAPSHOT フォルダにコピー <ol style="list-style-type: none"> g) パスから日付情報を取得した文字列を作成する。 Snapshot ファイルの“{USB 対象フォルダ}¥03062025110721_4095”部分を取り除く。 (例) 2025¥03¥06¥14¥01_43360_0300_01c0d.snap h) EventLog 側へコピーするときのファイル名を作成する。 上記 a) のパスを“¥”で区切った配列 index=1,2,3,4 を連結する。(yyyyMMddHH が期待値) その文字列に Snapshot ファイル名を“_”で連結する。 (例) 03061401_43360_0300_01c0d.snap_01_43360_0300_01c0d.snap i) {yyyy}¥{mm}¥{dd}¥{HH}¥03061401_43360_0300_01c0d.snap_01_43360_0300_01c0d.snap をコピー 9) Train Makeup Informaiton ファイルコピー 10) Data フォルダへ保存 以下のフォルダを圧縮する。 {テンポラリ}¥EVENTLOG フォルダ {テンポラリ}¥2025 フォルダ ※期待値:{テンポラリ}¥SNAPSHOT フォルダ {テンポラリ}¥TRAINMAKEUP フォルダ <p>2. MDS Event Log での MDS Snapshot 表示判定</p> <ol style="list-style-type: none"> 1) zip ファイルを解凍して.snap ファイルの有無とファイル名書式チェック 書式: SNAPSHOT¥{yyyy}¥{mm}¥{dd}¥{HH}¥ {yyyyMMddHH}_{CarID}_{Sequence Counter}_{SubsystemCode}_{EventCode}.snap 2) ファイルがあれば EventLog の“SNAPSHOT”列に“Y”をセット | <p>1. MDS Event Log の USB Import 処理</p> <ol style="list-style-type: none"> 1) USB Import ディレクトリ選択 2) データがある対象フォルダを検索「MMDDyyyyHHmmss_Car¥EVENTLOG」 3) 対象フォルダにある consist_info_EventLog.csv から編成情報取得 4) EVENTLOG フォルダをテンポラリにコピー 5) SNAPSHOT フォルダをテンポラリへコピーする 6) MDS Snapshot 変換処理呼び出し ConvertTcmsSnapshot()の引数に Snapshot と EventLog の格納フォルダをセットする。 Snapshot フォルダ: {テンポラリ}¥4095¥MDSSnapshot EventLog フォルダ: {テンポラリ}¥4095¥MDSEventLog 7) Snapshot 変換(Snapshot ファイルがある同じ場所に.snap ファイルを作成) 8) Snapshot のディレクトリ構造を維持したまま EVENTLOG と並列にある SNAPSHOT フォルダにコピー <ol style="list-style-type: none"> a) パスから日付情報を取得した文字列を作成する。 Snapshot ファイルの“{テンポラリ}¥4095¥MDSSnapshot”部分を取り除く。 (例) SNAPSHOT¥2025¥03¥06¥14¥01_43360_0300_01c0d.snap b) EventLog 側へコピーするときのファイル名を作成する。 上記 a) のパスを“¥”で区切った配列 index=1,2,3,4 を連結する。(yyyyMMddHH が期待値) その文字列に Snapshot ファイル名を“_”で連結する。 (例) 2025030614_01_43360_0300_01c0d.snap c) SNAPSHOT¥{yyyy}¥{mm}¥{dd}¥{HH}¥2025030614_01_43360_0300_01c0d.snap をコピー 9) Train Makeup Informaiton ファイルコピー 10) Data フォルダへ保存 以下のフォルダを圧縮する。 {テンポラリ}¥EVENTLOG フォルダ {テンポラリ}¥SNAPSHOT フォルダ {テンポラリ}¥TRAINMAKEUP フォルダ <p>2. MDS Event Log での MDS Snapshot 表示判定</p> <ol style="list-style-type: none"> 1) zip ファイルを解凍して.snap ファイルの有無とファイル名書式チェック 書式: SNAPSHOT¥{yyyy}¥{mm}¥{dd}¥{HH}¥ {yyyyMMddHH}_{CarID}_{Sequence Counter}_{SubsystemCode}_{EventCode}.snap 2) ファイルがあれば EventLog の“SNAPSHOT”列に“Y”をセット | | |

| | | | MITSUBISHI ELECTRIC CORPORATION | | | | | | | |
|--|--|--|---------------------------------|--|--|--|--|--|----------------------|-------|
| | | | 改 定 CHANGE | | | | | | 文 書 番 号 DOCUMENT NO. | |
| | | | L | | | | | | | |
| | | | | | | | | | JHCDF22 | 13-28 |

No.146 Slave REC からのダウンロード

| | |
|--|--|
| 変更点管理番号 No. | 146 |
| ソース・ファイル名 | frmLogDataDownload.cs, TrainImageFlexGrid.cs, HostedRadioButton.cs |
| モジュール名 | MDS_R211PTE.exe |
| 修 正 内 容 | |
| REC 選択用の RadioButton を追加する。ダウンロード時の要求先を画面で選択した REC にする。 | |
| 修 正 前 | 修 正 後 |
| <p>1. カスタムコントロール</p> <p>1) TrainImageFlexGrid コントロール</p> <p>a) 行定義</p> <pre>public Row ImageRow { get; set; } public Row CarNoRow { get; set; } : public Row UnitIdxRow { get; set; }</pre> <p>b) Column スタイル設定</p> <p>全行に対して ONLINE/OFFLINE/Unknown のスタイルを適用する。</p> <p>c) RecSelRowVisible プロパティ</p> <p>なし</p> | <p>1. カスタムコントロール</p> <p>1) TrainImageFlexGrid コントロール</p> <p>a) 行定義</p> <pre>public Row ImageRow { get; set; } public Row CarNoRow { get; set; } : public Row UnitIdxRow { get; set; }</pre> <p>b) Column スタイル設定</p> <p>REC 選択行を使っている場合、ONLINE のときは Normal スタイルにする。ColDataType が bool のときは ONLINE スタイルにするとチェックボックスになるため、OFFLINE、Unknown 時はそのままスタイルを適用する。</p> <p>c) RecSelRowVisible プロパティ</p> <p>Visible 状態を追加する。</p> <p>d) Master/Slave REC RadioButton</p> <p>グリッドのどの Column にきてもいいように全 Column に RadioButton を生成する。画面ロード時に生成して、online/offline 切替時は Visible のみ設定する。</p> |
| <p>2) HostedRadioButton コントロール</p> <p>a) イベントハンドラ</p> <p>CheckChanged()</p> <p>b) プロパティ</p> <p>Visible</p> <p>Checked</p> <p>CollIndex</p> <p>RowIndex</p> | <p>2) HostedRadioButton コントロール</p> <p>a) イベントハンドラ</p> <p>CheckChanged()</p> <p>Click()</p> <p>b) プロパティ</p> <p>Visible</p> <p>Checked</p> <p>CollIndex</p> <p>RowIndex</p> <p>AutoCheck</p> <p>GroupName</p> <p>IsMaster</p> |

| | |
|---|---|
| <p>2. 画面表示処理</p> <ol style="list-style-type: none"> 1) 各イベントハンドラ登録 2) 編成情報取得 3) 画面レイアウト <ol style="list-style-type: none"> a) TrainImageFlexGrid 生成する。 b) グリッドのスタイルなど初期設定をする。 c) Car 行を追加する(データ種別ごとに個別設定) d) データ種別ごとに表示する Sub-system のチェックボックスや記録期間していのコントロールを配置する。 4) 編成情報更新 <ol style="list-style-type: none"> a) Car イメージを表示する。 b) Offline の Car をマスク表示する。 <ul style="list-style-type: none"> - Column スタイルを設定する。 c) REC 選択ボタンを表示しない。 5) REC 選択ボタンのチェック処理 <p>なし</p> <p>3. ダウンロード処理</p> <ol style="list-style-type: none"> 1) ダウンロード要求作成 <ol style="list-style-type: none"> a) Event Log Master REC へ要求する。 b) RTDM Master REC へ要求する。 c) その他 Master REC へ要求する。 2) ダウンロード 3) 保存 | <p>2. 画面表示処理</p> <ol style="list-style-type: none"> 1) 各イベントハンドラ登録 2) 編成情報取得 3) 画面レイアウト <ol style="list-style-type: none"> a) TrainImageFlexGrid 生成する。 b) グリッドのスタイルなど初期設定をする。 c) Car 行を追加する(データ種別ごとに個別設定) d) データ種別ごとに表示する Sub-system のチェックボックスや記録期間していのコントロールを配置する。 e) EventLog と RTDM の場合、REC 選択ボタンを表示する。 <ul style="list-style-type: none"> - HostedRadioButton を全 Column に生成する。 - AutoCheck=false にしておく。 4) 編成情報更新 <ol style="list-style-type: none"> a) Car イメージを表示する。 b) Offline の Car をマスク表示する。 <ul style="list-style-type: none"> - Column スタイルを設定する。 c) REC 選択ボタンを表示する。 <ul style="list-style-type: none"> - Master/Slave 情報を TrainImageFlexGrid.MasterRecRow から取得する。 - TrainImageFlexGrid.UnitIdxRow から編成を取得して HostedRadioButton.GroupName に設定する。 - Master 優先でチェックを付ける。 - Master または Slave の Column のとき HostedRadioButton を表示する。 - offline から online の復帰で編成内に Checked=true の HostedRadioButton がないときに Master 優先でチェックする。Master がなければ Slave にチェックする。 - HostedRadioButton を Cell にあわせて配置する。 - 画面表示時にロードした HostedRadioButton を複数回ロードしないようにフラグ管理する。 5) REC 選択ボタンのチェック処理 <p>RadioButton はコンテナにのっているものをグルーピングする仕様のため、1 つのグリッド上に貼っている RadioButton を 2 つのグループに分けることはできない。そのため AutoCheck プロパティを false にして手動でチェックを制御する。また、AutoCheck=false にすると CheckClicked イベントが発生しないため Click イベントを使う。</p> <p>3. ダウンロード処理</p> <ol style="list-style-type: none"> 1) ダウンロード要求作成 <ol style="list-style-type: none"> a) Event Log 画面で選択した REC へ要求する。 b) RTDM 画面で選択した REC へ要求する。MonGen 定義も同様。 c) その他 Master REC へ要求する。 2) ダウンロード 3) 保存 |
|---|---|

| | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|-----------------------------|--|
| | | | <p>MITSUBISHI ELECTRIC CORPORATION</p> <p>改 定 CHANGE</p> <p>文 書 番 号 DOCUMENT NO.</p> | | | | | | | |
| | | | L | | | | | | <p>JHCDF22</p> <p>13-30</p> | |