

BrbLibUa V1.03 – Revisionsgeschichte

Inhaltsverzeichnis

1 Revisionsgeschichte	2
1.1 BrbLibUa V1.03 – 2024-02-22	2
1.1.1 Hinweise zum Compiler	2
1.1.2 Code-Prüfung mit ClangTidy	2
1.1.3 Versionen geändert	2
1.1.4 Abhängigkeit geändert	2
1.2 BrbLibUa V1.02 – 2023-06-07	3
1.2.1 Versionen geändert	3
1.2.2 Abhängigkeit geändert	3
1.2.3 Implementierung von Unit-Tests	3
1.2.4 Code-Prüfung mit ClangTidy	3
1.2.5 Neues Paket ‚General - ServerInfo‘	3
1.2.6 Neues Paket ‚General - ServerDiag‘	3
1.2.7 Umbenennung von GetText-Funktionen	3
1.2.8 Verwendung der sicheren String-Funktionen	4
1.2.9 Optimierung einiger Funktionen: Null-Pointer-Prüfung	4
1.2.10 Erweiterung bei ‚BrbUaConvVariantValueFromString‘	4
1.2.11 Erweiterung bei ‚BrbUaConvVariantValueToString‘	4
1.2.12 Neue Funktion ‚BrbUaAddLocalizedTextText‘	4
1.2.13 Erweiterung bei ‚BrbUaGetRandomGuidString‘	4
1.2.14 RunClient: Korrektur beim Parameter ‚ApplicationName‘	4
1.2.15 Neuer Anhang für dynamische Arrays	5
1.2.16 Neuer Anhang für 64-Bit-Datentypen	5
1.3 BrbLibUa V1.01 – 2022-05-13	6
1.3.1 Hinweise auf GitHub	6
1.3.2 Neues Paket ‚TimeAndDate‘	6
1.3.3 RunClient: Korrektur beim Einlesen von Ausgangs-Argumenten	6
1.3.4 RunClient: Korrektur beim Einlesen von Argumenten	6
1.3.5 RunClient: Korrektur beim Anmelden eines Event-Fields	6
1.3.6 RunClient: Korrektur beim Empfangen eines Events	6
1.3.7 RunClient: Optimierung des Monitors	6
1.3.8 RunServer: Korrektur beim Mappen eines Event-Fields	6
1.3.9 RunServer: Optimierung des Monitors	7
1.4 BrbLibUa V1.00 – 2021-09-14	7

1 Revisionsgeschichte

Ab V1.02 ist die Revisionsgeschichte aus der Hilfe in diese Datei extrahiert worden.

1.1 BrbLibUa V1.03 – 2024-02-22

1.1.1 Hinweise zum Compiler

In diese Hilfe wurden die [Hinweise zum Compiler](#) aufgenommen.

1.1.2 Code-Prüfung mit ClangTidy

Die Prüfung mit ClangTidy des Entwicklungs- und Demo-Projekts (siehe Allgemeines/ [Geprüft mit ClangTidy](#)) wurde mit einer neuen Version durchgeführt (die vorige Version prüfte manche Tasks nicht komplett). Die dadurch erkannten Code-Stellen wurden überprüft und gegebenenfalls optimiert. Die Bibliothek wurde dabei nicht geändert.

1.1.3 Versionen geändert

Folgende Versionen wurden im Projekt geändert:

- AR von E4.91 auf H4.91

Auf die Funktion und die Kompatibilität hat dies keine Auswirkung.

1.1.4 Abhängigkeit geändert

Die Abhängigkeit von der Basis-Bibliothek „BrbLib“ wurde von V5.02 auf V5.03 geändert.

1.2 BrbLibUa V1.02 – 2023-06-07

Ab V1.02 ist hier nur die letzte Version erwähnt. Die gesamte Revisionsgeschichte wurde in die Datei „BrbLibUa Revisionsgeschichte“ ausgelagert.

1.2.1 Versionen geändert

Folgende Versionen wurden im Projekt geändert:

- Compiler von V4.1.2 auf 6.3.0
- AR von D4.91 auf E4.91

Auf die Funktion und die Kompatibilität hat dies keine Auswirkung.

1.2.2 Abhängigkeit geändert

Die Abhängigkeit von der Basis-Bibliothek „BrbLib“ wurde von V5.01 auf V5.02 geändert.

1.2.3 Implementierung von Unit-Tests

Mithilfe der AS-Solution ‚UnitTest‘ (Unterstützung für wiederholbare Aufruf-Tests und deren Berichte) wurden Test-Aufrufe für sehr viele Funktionen mit den unterschiedlichsten Parametern (Null-Pointer, ungültige Parameter, Grenzen usw.) implementiert.

Damit wird eine erhebliche Verbesserung der Software-Qualität erreicht. So wurden einige Fehlverhalten mancher Funktionen (meist bei grenzwertigen Parametern) erkannt und behoben. Siehe Allgemeines/[Getestet mit UnitTests](#).

1.2.4 Code-Prüfung mit ClangTidy

Das gesamte Entwicklungs- und Demo-Projekt wurde mit dem Code-Analyse-Tool ClangTidy geprüft, welches zur Erhöhung der Software-Qualität beiträgt. Genauere Details dazu siehe den neuen Abschnitt Allgemeines/[Geprüft mit ClangTidy](#).

Einige der folgenden Änderungen basieren auf dem von ClangTidy erstellten Bericht.

Manche Funktionen wurden dadurch korrigiert, die meisten davon sollten das Verhalten nicht beeinflussen. Somit bleibt diese Version kompatibel zur Vorgänger-Version.

1.2.5 Neues Paket ‚General - ServerInfo‘

Das Paket enthält keine Funktionen, sondern nur Datentyp-Definitionen. Diese können verwendet werden, um Info-Daten des Servers zu lesen oder zu abonnieren (z.B. die aktuelle Serverzeit oder die Software-Version). Siehe [ServerInfo](#).

1.2.6 Neues Paket ‚General - ServerDiag‘

Das Paket enthält keine Funktionen, sondern nur Datentyp-Definitionen. Diese können verwendet werden, um Diagnose-Daten des Servers zu lesen oder zu abonnieren (z.B. wie viele und welche Clients verbunden sind). Siehe [ServerDiag](#).

1.2.7 Umbenennung von GetText-Funktionen

Bei einigen Funktionen, die einen Wert in Klartext wandeln, wird der übergebene String zur Aufnahme des Klartextes nicht gelöscht, sondern der Klartext wird nur angehängt, weil dies für den Anwender meistens bequemer ist.

Wird eine solche Funktion also zyklisch aufgerufen, ohne jeweils vorher den String zu löschen, kommt es zu einem **Speicherschmierer**.

Da dies ungeahnte Folgen haben kann, wurden die betroffenen Funktionen in ‚Add‘ umbenannt und deren Beschreibung in dieser Hilfe verbessert, um auf diesen Umstand explizit hinzuweisen.

Folgende Funktionen sind betroffen:

Alter Name	Neuer Name
BrbUaGetNodeIdText	BrbUaAddNodeIdText
BrbUaGetBooleanText	BrbUaAddBooleanText
BrbUaGetNodeClassText	BrbUaAddNodeClassText
BrbUaGetDatatypeIdText	BrbUaAddDatatypeIdText
BrbUaGetArrayDimensionText	BrbUaAddArrayDimensionText
BrbUaGetAccessLevelText	BrbUaAddAccessLevelText
BrbUaGetEventNotifierText	BrbUaAddEventNotifierText

Achtung: Da sich die Namen der Funktionen geändert haben, müssen auch etwaige Aufrufe geändert werden!

1.2.8 Verwendung der sicheren String-Funktionen

Bei Funktionen, bei denen es ohne Kompatibilitätsverlust möglich war, wurden ‚strcpy‘ durch ‚BrbStringCopy‘ und ‚strcat‘ durch ‚BrbStringCat‘ ersetzt. Diese erst kürzlich implementierten BrbString-Funktionen garantieren ein sicheres Kopieren/Anhängen eines Strings (siehe Dokumentation der Basis-Bibliothek „BrbLib“). Dadurch werden auch die Ua-Funktionen sicherer.

Folgende Funktionen sind betroffen:

```
BrbUaGetRandomNodeId  
BrbUaGetAttributeList  
BrbUaSetLocalizedText  
BrbUaGetRandomLocalizedText  
BrbUaSetQualifiedName  
BrbUaGetRandomXmlElement  
BrbUaGetConnectionStatusText  
BrbUaSrvGetMethodOperateText
```

Achtung: Dadurch ändert sich die Abhängigkeit zur Version der Basis-Bibliothek „BrbLib“ von V5.01 auf V5.02.

Der Aufruf der sichereren String-Funktionen bedingt die Übergabe der Ziel-String-Größe. In Funktionen, welche den Zielstring lediglich als Zeiger zur Verfügung haben, musste daher die Signatur dahingehend geändert werden, dass auch die String-Größe als Argument übergeben wird.

Folgende Funktionen sind betroffen:

```
BrbUaAddNodeIdText  
BrbUaAddBooleanText  
BrbUaAddNodeClassText  
BrbUaAddDatatypeIdText  
BrbUaAddArrayDimensionText  
BrbUaAddAccessLevelText  
BrbUaAddEventNotifierText  
BrbUaAddVariantValueSubName  
BrbUaAddLocalizedTextText  
BrbUaGetRandomGuidIdString
```

Achtung: Da sich die Argument-Anzahl dieser Funktionen geändert hat, müssen auch etwaige Aufrufe geändert werden!

1.2.9 Optimierung einiger Funktionen: Null-Pointer-Prüfung

In folgenden Funktionen werden jetzt auch Null-Pointer geprüft:

```
BrbUaAreNodeIdsEqual, BrbUaAddArrayDimensionText
```

1.2.10 Erweiterung bei ‚BrbUaConvVariantValueFromString‘

Bei der Funktion [BrbUaConvVariantValueFromString](#) wird jetzt auch der Datentyp `NodeId` unterstützt. Möglich sind hier die Identifier-Typen `Numeric`, `String` und `Guid`.

Auch der Datentyp `Guid` wird jetzt unterstützt. Er wird im Element `String` eingetragen.

Zusätzlich unterstützt wird der Datentyp `StatusCode`. Er wird im Element `UInt32` eingetragen. Zugelassen als Text sind Dezimalzahlen und Hex-Zahlen mit dem Präfix `0x`.

1.2.11 Erweiterung bei ‚BrbUaConvVariantValueToString‘

Bei der Funktion [BrbUaConvVariantValueToString](#) wird jetzt auch der Datentyp `NodeId` unterstützt. Möglich sind hier die Identifier-Typen `Numeric`, `String` und `Guid`.

Auch der Datentyp `Guid` wird jetzt unterstützt. Der Wert wird aus dem Element `String` genommen.

1.2.12 Neue Funktion ‚BrbUaAddLocalizedTextText‘

Die Funktion wandelt einen `LocalizedText` in einen Klartext um und hängt diesen an den übergebenen `String` an.

So können `LocalizedTexts` bequem visualisiert werden.

1.2.13 Erweiterung bei ‚BrbUaGetRandomGuidIdString‘

Bei der Funktion [BrbUaGetRandomGuidIdString](#) kann durch ein neues Eingangs-Argument festgelegt werden, ob das Trennzeichen „-“ eingefügt wird oder nicht.

Achtung: Da sich die Argument-Anzahl dieser Funktion geändert hat, müssen auch etwaige Aufrufe geändert werden!

1.2.14 RunClient: Korrektur beim Parameter ‚ApplicationName‘

Der bei der Anmeldung des Clients verwendete Parameter ‚SessionConnectInfo.ApplicationName‘ wurde nicht aus dem Parameter ‚AppName‘ des Datenmoduls übernommen, sondern leer gelassen.

1.2.15 Neuer Anhang für dynamische Arrays

Unter [Hinweise zu dynamischen Arrays](#) wird erklärt, wie dynamische Arrays funktionieren und eingebunden werden können.

1.2.16 Neuer Anhang für 64-Bit-Datentypen

Unter [Hinweise zu 64-Bit-Datentypen](#) wird erklärt, wie 64-Bit-Datentypen (UInt64 und Int64) eingebunden werden können.

1.3 BrbLibUa V1.01 – 2022-05-13

1.3.1 Hinweise auf GitHub

In diese Hilfe wurden die Hinweise auf die GitHub-Links aufgenommen.

1.3.2 Neues Paket ‚TimeAndDate‘

Es enthält Funktionen für Zeit und Zeitzonen, z.B. zur Umwandlung des OpcUa-Datentyps ‚UaSrv_DateTimeType‘ in andere Zeit-Datentypen und zurück oder zur Umwandlung in ein Textformat.

1.3.3 RunClient: Korrektur beim Einlesen von Ausgangs-Argumenten

Die Anzahl der Ausgangs-Argumente einer Methode ist vom Plc-Open-Funktionsblock aus der Client-Bibliothek auf 10 beschränkt. Diese Limitierung wurde nicht überprüft. Sind bei einer Methode im Datenobjekt mehr als 10 Ausgangs-Argumente angegeben, kann es zu einer Speicherverletzung führen.

1.3.4 RunClient: Korrektur beim Einlesen von Argumenten

Wurde kein Write-Block angegeben, wurden zwar alle Methoden aus dem Datenobjekt eingelesen, das Einlesen der dazugehörigen Argumente wurde jedoch übersprungen. Beim Aufruf einer Methode mit Argumenten wurde dann der Fehler ‚0x80AB0000 = Bad_InvalidArgument‘ zurückgegeben.

1.3.5 RunClient: Korrektur beim Anmelden eines Event-Fields

Im Datenobjekt wird bei einem Event-Field der Namespace-Index des Datenobjekts angegeben. Dieser wurde aber nicht mit dem zur Laufzeit gültigen Index ersetzt. Das Event wurde deshalb nicht empfangen. Stattdessen wurde beim Event-Item der Fehler

‚0xB0080000 = PlcOpen_BadEventFieldSelection‘ eingetragen.

Hinweis: Bei von der Opc-Foundation definierten Event-Fields fällt dieser Fehler nicht auf, weil sowohl im Datenobjekt als auch zur Laufzeit der Index 0 ist.

Achtung: Es war bisher möglich, diesen Fehler zu umgehen, indem hier die zur Laufzeit gültigen Indizes angegeben wurden. Sollte dies in einer bestehenden Applikation der Fall sein, so müssen diese auf die Namespace-Indizes des Datenmoduls geändert werden!

1.3.6 RunClient: Korrektur beim Empfangen eines Events

Ist für ein Event-Item der Parameter ‚CallOperate‘ auf 1 gesetzt, so wird das Empfangen eines Events vom Cyclic-FB behandelt. Unter anderem wird dadurch das Flag ‚bEventReceived‘ beim Empfang auf 1 gesetzt. Es sollte eigentlich nur 1 Zyklus lang gesetzt bleiben. Bis jetzt konnte das Flag aber mehrere Zyklen gesetzt bleiben. So wurde beim zyklischen Aufruf der Hilfs-Funktion ‚BrbUaRcGetEventItemReceived‘ der Anschein erweckt, dass das Event mehrmals empfangen wurde.

Jetzt wird das Flag für jedes Event-Item in jedem Zyklus gelöscht.

1.3.7 RunClient: Optimierung des Monitors

Ist der Monitor nicht aktiviert, so zeigt jetzt nicht nur der Gesamt-Monitor-Status, sondern auch der Monitor-Status jeden Elements den Wert ‚eBRB_ERR_NOT_ENABLED=50002‘.

1.3.8 RunServer: Korrektur beim Mappen eines Event-Fields

Im Datenobjekt wird ein Event-Field eines Events über den Browse-Pfad gemappt (z.B. „Name=/Severity“ oder „Name=/Transition/Id“). Dieser kann aus einem oder mehreren Elementen bestehen, welche durch / getrennt sind. Jedes Element beinhaltet einen Namespace-Index. Bei von der Opc-Foundation definierten Events ist dieser immer 0 und kann auch weggelassen werden. Bei Events aus Fremdmodellen ist er größer 0 (z.B. „Name=/7:Error/7:Text“) und kann sich zur Laufzeit verschieben. Bis jetzt wurde dieser Index nicht ersetzt.

Ab dieser Version werden alle Namespace-Indizes aller Browse-Pfade durch die zur Laufzeit geltenden Indizes ersetzt. So können auch Event-Fields von Fremdmodellen korrekt gemappt werden.

Achtung: Es war auch schon bisher möglich, Browse-Pfade mit Namespace-Indizes zu versehen. Allerdings wurden diese vom RunServer nicht ersetzt, so dass hier die zur Laufzeit gültigen Indizes angegeben werden mussten. Sollte dies in einer bestehenden Applikation der Fall sein, so müssen diese auf die Namespace-Indizes des Datenmoduls geändert werden!

1.3.9 RunServer: Optimierung des Monitors

Ist der Monitor nicht aktiviert, so zeigt jetzt nicht nur der Gesamt-Monitor-Status, sondern auch der Monitor-Status jeden Elements den Wert ‚eBRB_ERR_NOT_ENABLED=50002‘.

1.4 BrbLibUa V1.00 – 2021-09-14

Die Bibliothek wurde neu erstellt.