

RnCommTest – Revisionsgeschichte

Inhaltsverzeichnis

1 Revisionsgeschichte	8
1.1 RnCommTest V8.00	8
1.1.1 Allgemein	8
1.1.1.1 Umstellung des Dot.Net-Frameworks auf 4.8.....	8
1.1.1.2 Neue Haupt-Tab-Leiste	8
1.1.2 OpcUa	8
1.1.2.1 Umstellung des OpcUa-Stacks auf 1.05.374.54.....	8
1.1.2.2 Einheitliches Verzeichnis für alle PubSub-Konfigurationen	8
1.1.2.3 OpcUaPubSubConfigurator.....	8
1.1.2.3.1 Erweiterung des Konfigurations-Typen	8
1.1.2.3.2 Erweiterung um Parameter ‚SecurityMode‘	8
1.1.2.3.3 Erweiterung um Datentyp UInt64 für PublisherId laut V1.05	8
1.1.2.3.4 Datentyp-Prüfung bei DragAndDrop optimiert.....	9
1.1.2.4 OpcUaClient.....	9
1.1.2.4.1 Korrektur wegen AR6.00	9
1.1.2.4.2 Verändern der Benutzer-Reihenfolge in der Anmelde-Liste	9
1.1.2.4.3 Korrekturen beim Akzeptieren von Server-Zertifikaten.....	9
1.1.2.4.4 Neue Such-Funktion FindNodeId	9
1.1.2.4.5 Optimierung bei Browse-Icons	9
1.1.2.4.6 Optimierung/Erweiterung bei der Anzeige/Editierung von Attributen.....	9
1.1.2.4.7 Erweiterung beim Auflösen von Datentypen	10
1.1.2.4.8 Optimierung bei Event-Filter-Anzeige.....	10
1.1.2.4.9 Optimierung bei ServerDiag	10
1.1.2.5 OpcUaServer	10
1.1.2.5.1 Anpassungen für neuen Stack wegen RolePermissions	10
1.1.2.5.2 Korrekturen beim Laden von Fremdmodellen	10
1.1.2.6 OpcUaBrMapper	10
1.1.2.6.1 Erweiterung um das uad-Format von AS6.00	10
1.1.2.7 Neues Modul OpcUaDevConfigurator.....	10
1.2 RnCommTest V7.06	13
1.2.1 Allgemein	13
1.2.1.1 Neu: Versions-Informationen verwendeter Software	13
1.2.1.2 Neues Modul „TextFormatter“	13
1.2.1.3 Text-Fenster	13
1.2.2 OpcUa	13
1.2.2.1 Korrektur am OpcUa-Stack	13
1.2.2.2 OpcUaClient.....	13
1.2.2.2.1 Optimierung bei der Sortierung von Sub-Knoten.....	13
1.2.2.2.2 Korrektur beim Suchen.....	13
1.2.2.2.3 Erweiterung für WriteOnly-Variablen.....	13
1.2.2.2.4 Neuer Menüpunkt im Browse-Fenster: Call method without arguments.....	14
1.2.2.2.5 Neuer Menüpunkt im Browse-Fenster: Add all Methods	14
1.2.2.2.6 Optimierung beim Auflösen eines Datentyps	14
1.2.2.2.7 Korrektur im Reiter Attributes/Advanced	14
1.2.2.2.8 Erweiterung im Reiter MethodCall.....	14
1.2.2.2.9 Dialogfeld für Methoden-Argumente (Show in dialog)	14
1.2.2.3 OpcUaSubscriber.....	14
1.2.2.3.1 Optimierung der Wert-Anzeige.....	14
1.2.2.4 OpcUaServer	14
1.2.2.4.1 Änderung beim Standard-Wert für EngineeringUnit	14
1.2.2.4.2 Object-Knoten mit EventNotifier	14
1.2.2.4.3 Feuern von Events an eigenen und externen Object-Knoten.....	14
1.2.2.4.4 Erweiterung für Enumerations	14
1.2.2.4.5 Korrektur am Method-Code-Fenster.....	15
1.2.2.4.6 Neues Kommando bei Method-Code: Feuern eines Events	15
1.2.2.4.7 Neues Kommando bei Method-Code: Setzen des Rückgabe-Status	15
1.2.2.4.8 Erweiterungen und Optimierungen für externe Modelle	15
1.2.2.4.9 Fehlerbehebung bei BrowseNames	15
1.2.2.5 OpcUaBrMapper	16
1.2.2.5.1 Erweiterung beim Einlesen eines Fremdmodells	16
1.2.2.5.2 Erweiterung beim Einlesen einer DefaultView für AS6.00.....	16

1.3 RnCommTest V7.05	17
1.3.1 OpcUa - PubSubConfigurator	17
1.3.1.1 Korrektur beim Import einer Datei mit mehreren Konfigurationen	17
1.3.1.2 Optimierung für GroupVersion	17
1.3.1.3 Anzeige der Properties	17
1.3.2 OpcUaClient	17
1.3.2.1 Optimierung bei Anzeige einer Enumeration	17
1.3.2.2 Erweiterung bei Wert-Editor	17
1.3.2.3 Korrekturen und Erweiterungen für TemporaryFileType	17
1.3.2.4 MethodCall: Erweiterungen und Optimierungen	18
1.3.3 OpcUaSubscriber	18
1.3.3.1 Anzeige einer Statistik aller empfangenen Werte	18
1.3.3.2 Sortierung der empfangenen Werte	18
1.3.4 OpcUaServer	18
1.3.4.1 Optimierungen für FileType	18
1.3.4.2 AddressSpace: Erweiterung um TemporaryFileTransferType	18
1.3.5 OpcUaBrMapper	18
1.3.5.1 Korrekturen beim Einlesen des Fremdmodells	18
1.4 RnCommTest V7.04	19
1.4.1 OpcUa - PubSub	19
1.4.1.1 Neuer Editor für PubSub-Konfigurationen	19
1.4.2 OpcUaClient	19
1.4.2.1 Korrektur und Optimierung für FileTransfer	19
1.4.2.2 OpcUaClient – Attribut-Liste: Korrektur bei Wert-Anzeige als Dialog	19
1.4.2.3 MethodCall: Optimierung der Methoden-Parameter-Anzeige	19
1.4.2.4 MethodCall: Optimierung für abgeleitete Datentypen	19
1.4.2.5 MethodCall: Korrektur bei der Anzeige von Ausgangs-Argumenten	19
1.4.2.6 MethodCall: Korrektur bei der Anzeige von Array-Argumenten	19
1.4.2.7 MethodCall: Erweiterung für Argumente mit Datentyp „Guid“	19
1.4.2.8 Neuer Reiter PubSubConfigurator	19
1.4.3 OpcUaSubscriber	19
1.4.3.1 OpcUaSubscriber: Anpassung an den neuen Editor	19
1.4.4 OpcUaServer	20
1.4.4.1 Neue Funktion: Laden von externen Modellen	20
1.4.4.2 Adressraum: Anzeige von Types-Knoten	20
1.4.4.3 Unterstützung für ComplexTypeServerFacet2017	20
1.4.4.4 Methoden-Interpreter: Erweiterung um String-Literale mit Leerzeichen	20
1.4.4.5 Erweiterungen um Datentyp „Guid“	20
1.4.4.6 File: Neuer Parameter ‚Access timeout‘	20
1.4.4.7 Publisher: Anpassung an den neuen Editor	20
1.4.4.8 Beispiel-Konfiguration um Datei erweitert	21
1.4.5 OpcUaBrMapper	21
1.4.5.1 Korrektur bei „Clear Settings“	21
1.4.5.2 Optimierung für Präfixes	21
1.5 RnCommTest V7.03	22
1.5.1 Allgemein	22
1.5.1.1 Verbesserung der Hilfe durch Links	22
1.5.1.2 Hinweis auf GitHub aufgenommen	22
1.5.1.3 Allgemein: Automatisches Speichern beim Beenden entfernt	22
1.5.1.4 Allgemein: Rücksetzen des Layouts	22
1.5.1.5 Allgemein: Optimierung Trend	22
1.5.2 OpcUa	22
1.5.2.1 Neues Modul „OpcUaBrMapper“	22
1.5.2.2 Umstellung des OpcUa-Stacks	22
1.5.3 OpcUaClient	22
1.5.3.1 OpcUaClient: Optimierung für falschen Attribut-Status	22
1.5.3.2 OpcUaClient: Unterstützung für ComplexTypeServerFacet2017	23
1.5.3.3 OpcUaClient: Optimierung beim Auflösen eines Datentyps	23
1.5.3.4 OpcUaClient – Attribut-Liste: Wert-Anzeige als Dialog	23
1.5.3.5 OpcUaClient: Unterstützung des File-Transfers	23
1.5.3.6 OpcUaClient: Optimierung der Node-Browse-Dialogs	23
1.5.3.7 OpcUa-Client – Access.Advanced: Erweiterung für OpcUa-Datentyp	24
1.5.3.8 OpcUaClient – Subscription: Optimierung der Settings-Dialoge	24
1.5.3.9 OpcUaClient – Subscription – Event-Filter: Erweiterung der Field-Auswahl	24
1.5.3.10 OpcUaClient – Method Call: Korrektur für BaseDataType	24
1.5.3.11 OpcUaClient – Method Call: Erweiterung der Datentypen für Argumente	24
1.5.3.12 OpcUaClient – Method Call: Speichern und Laden der Methoden-Liste	24
1.5.3.13 OpcUaClient – Method Call: Ändern der Description in der Liste	24
1.5.3.14 OpcUaClient – Method Call: Sequentieller Aufruf der Liste	24
1.5.3.15 OpcUaClient – Method Call: Setzen des FileHandles bei Datei-Methoden	24

1.5.3.16 OpcUaClient – Method Call: Nachträgliches Einsehen der Argumente pro Aufruf	25
1.5.3.17 OpcUaClient – OpcUaMonitor: Speichern der Filter-Einstellungen	25
1.5.4 OpcUaSubscriber	25
1.5.4.1 OpcUaSubscriber: Import mehrerer Konfigurationen möglich	25
1.5.4.2 OpcUaSubscriber: Optimierung Stop-Funktion	25
1.5.4.3 OpcUaSubscriber: Optimierung der Ausgabe mit neuen Optionen	25
1.5.4.4 OpcUaSubscriber: Neues Dialogfeld „Statistics“	25
1.5.4.5 OpcUaSubscriber: Korrektur am Stack	25
1.5.4.6 OpcUaSubscriber: Korrektur/Optimierung beim Trace	25
1.5.5 OpcUsServer	25
1.5.5.1 OpcUaServer - AddressSpace: Erweiterung um Files.....	25
1.6 RnCommTest V7.02	27
1.6.1 Pvi – Update auf PviServices 4.10.1.42	27
1.6.2 ModbusTcpMaster: Optimierung der Schreib-Ausgabe	27
1.6.3 Zertifikats-Verwaltung: Erweiterung für Benutzer-Zertifikate	27
1.6.4 OpcUaClient – Server-Properties: Interner Umbau der Einstellungen	27
1.6.5 OpcUaClient – Connection: Optimierung und Erweiterung der Benutzer-Authentifizierungs-Daten-Verwaltung	27
1.6.6 OpcUaClient – Connection: Erweiterung um Anmeldung mit Zertifikat	27
1.6.7 OpcUaClient – Connection: Verwaltung der bevorzugten Sprachen.....	27
1.6.8 OpcUaClient – Connection: Auswahl des Client-Instanz-Zertifikats	27
1.6.9 OpcUaClient – Connection: Benutzer-Änderung während einer Verbindung	27
1.6.10 OpcUaClient – MethodCall: Asynchroner Aufruf möglich.....	27
1.6.11 OpcUaClient – ServerDiag: Ausführliche Anzeige komplexer Werte	28
1.6.12 OpcUaClient – ServerDiag: Trace der Security-Werte möglich.....	28
1.6.13 OpcUaClient – Monitor: Update der Session möglich	28
1.6.14 OpcUaSubscriber – Diag: GroupVersion zusätzlich als Zeitstempel.....	28
1.6.15 OpcUaServer – UserManagement: Erweiterung um Anmeldung mit Zertifikat	28
1.6.16 OpcUaServer – Ausgabefenster: Ausgaben für Clients	28
1.7 RnCommTest V7.01	29
1.7.1 Optimierung dieses Hilfe-Dokuments.....	29
1.7.2 Zertifikate: Neue Hinweise zur Verwendung bei B&R	29
1.7.3 Umstellung des OpcUa-Stacks	29
1.7.4 OpcUa-Server: Korrektur bei Zertifikats-Prüfung zur Anmeldung verschlüsselter Verbindungen	29
1.7.5 OpcUa-Client: Optimierung am Layout	30
1.7.6 OpcUaClient – Access: Neuer Reiter „RegisterNodes“	30
1.7.7 OpcUa-Subscriber: Korrektur bei Datentypen ‚String‘ und ‚ByteString‘	30
1.7.8 OpcUa-Server/Publisher: Korrektur bei Datentypen ‚String‘ und ‚ByteString‘	30
1.7.9 OpcUa-Server/Publisher: Korrektur bei Datentyp ‚String‘	30
1.7.10 OpcUa-Server/Publisher: Korrektur bei Datentyp ‚DateTime‘	30
1.7.11 OpcUa-Subscriber: Erweiterung bei Diagnose.....	30
1.7.12 ModbusTcpSlave: Optimierung der Ausgabe.....	30
1.7.13 ModbusTcpSlave: Korrektur beim Auswerten von Telegrammen	30
1.8 RnCommTest V7.00	31
1.8.1 Ethernet: Erweiterung beim Auslesen der Adapter	31
1.8.2 Umstellung des OpcUa-Stacks	31
1.8.3 Neues Modul OpcUa-Subscriber	31
1.8.4 OpcUa-Server: Neuer Reiter ‚Publisher‘	31
1.8.5 OpcUa-Server: Neue Funktionalität ‚ReverseConnect‘	31
1.8.6 OpcUa-Client: Neue Funktionalität ‚ReverseConnect‘	31
1.8.7 OpcUa-Client: Korrektur bei Ermittlung von Struktur-Definitionen.....	31
1.8.8 OpcUa-Client: Optimierung beim Auslesen von Attributen.....	31
1.8.9 OpcUa-Client: Korrektur bei zurückgewiesenden Zertifikaten	31
1.8.10 Pvi – Korrektur bzw. Erweiterung beim Service-Browsing	31
1.8.11 Pvi – Veränderbare Item-Größe	32
1.8.12 Pvi – Neue Einstellung ‚Log ValueChange events‘	32
1.8.13 Pvi – Neue Angaben beim Object ‚Service‘	32
1.8.14 Pvi – Mehr Cpu-Informationen	32
1.8.15 Pvi – Optimierung bei Logbuch	32
1.8.16 Pvi – Neue Angaben beim Object ‚Hardware‘	32
1.8.17 Pvi – Korrektur bei der Trendanzeige.....	32
1.9 RnCommTest V6.01	33
1.9.1 Trend: Korrektur	33
1.9.2 Udp: Erweiterungen	33
1.9.3 Umstellung des OpcUa-Stacks	33
1.9.4 OpcUa – Zertifikats-Erstellung: Korrektur + neue Angaben	33
1.9.5 OpcUa – Zertifikats-Verwaltung: Neue Funktion ‚Show certificate from hex string‘	33
1.9.6 OpcUa-Client: Optimierung bei Knoten-Browse-Dialog.....	33
1.9.7 OpcUaClient – Server-Properties: Einstellung ‚Replace discovered hostname with IP‘	33
1.9.8 OpcUaClient – Server-Properties: Geänderte Einstellung ‚Browse sort mode‘	34

1.9.9 OpcUaClient – Server-Properties: Neue Einstellung ‚Clear subscription lists on connect‘	34
1.9.10 OpcUaClient – Server-Properties: Neue Einstellungen für EventItems	34
1.9.11 OpcUaClient – Server-Properties: Neue Einstellung ‚Do not disconnect when tool is closed‘	34
1.9.12 OpcUaClient – Server-Properties: Neue Einstellung ‚Advanced – Trim slash at end of endpoint URL‘	34
1.9.13 OpcUa-Client – Connection: Optimierung der Fehlerausgabe bei Verbindungsaufbau	34
1.9.14 OpcUaClient – Connection: Optimierung beim Aufbauen einer Verbindung	35
1.9.15 OpcUaClient – Server-Properties: Korrektur bei Änderung des Benutzers	35
1.9.16 OpcUa-Client – Access.Advanced: Anzeige von Byte() als Zertifikat	35
1.9.17 OpcUa-Client – Access.Advanced: Korrektur bei ExtensionObjects	35
1.9.18 OpcUa-Client – Access.Advanced: Optimierung bei Enumerationen	35
1.9.19 OpcUa-Client – Access.Advanced: Anzeige von String-Collections	35
1.9.20 OpcUaClient - Subscription: Korrektur bei der Trendanzeige	35
1.9.21 OpcUaClient - Subscription: Implementierung für Alarms & Conditions	35
1.9.22 OpcUaClient - Subscription: Auflösung von Event-Feldern mit komplexem Datentyp	36
1.9.23 OpcUaClient - Subscription: Erweiterung beim automatischen Abonnieren von Event-Feldern	36
1.9.24 OpcUaClient - MethodCall: Unterstützung weiterer OpcUa-Datentypen	36
1.9.25 OpcUaClient – Historical Access: Neue Fehlerausgabe	36
1.9.26 OpcUaClient – ServerDiag: Korrektur bei Stop	36
1.9.27 OpcUaClient - Monitor: Optimierung bei Event-Items	36
1.9.28 OpcUaServer – Settings: Neuer Parameter ‚Correct client signature algorithm‘	36
1.9.29 OpcUaServer – Diag: Ausgabe des Client-Zertifikats	36
1.10 RnCommTest V6.00	37
1.10.1 Umstellung der Entwicklungsumgebung MS Visual Studio	37
1.10.2 Umstellung des Dot.Net-Frameworks	37
1.10.3 Layout-Anpassungen für Windows10	37
1.10.4 Umstellung des OpcUa-Stacks	37
1.10.5 OpcUaClient/Server: Unterstützung neuer Security-Policies	37
1.10.6 OpcUaClient – Server-Properties: Neue Einstellung ‚Evaluate sub nodes on browsing‘	37
1.10.7 OpcUaClient – Server-Properties: Neue Einstellung ‚Browse sort mode‘	37
1.10.8 OpcUaClient: Optimierung bei Auflösung von Datentypen	37
1.10.9 OpcUaClient: Überarbeitung des Wert-Steuerlements	37
1.10.10 OpcUaClient - Subscription: Anzeigen des Wertes in eigenem Dialog	38
1.10.11 OpcUaClient – MethodCall: Korrektur bei dynamischen Array-Argument	38
1.10.12 OpcUaClient – MethodCall: Korrektur bei String-Array-Argument	38
1.10.13 OpcUaClient – Certificates: Längerer Gültigkeitsbereich möglich	38
1.11 RnCommTest V5.04	39
1.11.1 Layout-Anpassungen für Windows10	39
1.11.2 Neuer Dialog zum Anzeigen/Verändern von Grafiken (Image)	39
1.11.3 Ethernet	39
1.11.4 ModBusTcp Slave – Erweiterte Ausgabe der Daten	39
1.11.5 Pvi – Korrektur beim Anlegen einer ANSL-Verbindung	39
1.11.6 OpcUaClient/Server – Korrektur bei Eingabe von Datums-Werten	39
1.11.7 OpcUaClient - Endpoint: Optimierung des Fensters	39
1.11.8 OpcUaClient – Endpoint - Search: Anpassung der Adapter-Liste	39
1.11.9 OpcUaClient - Browse: Korrektur beim Suchen	39
1.11.10 OpcUaClient – Browsen bei Eingabe einer NodeId	39
1.11.11 OpcUaClient – Speichern und Laden von Werten/Argumenten: Neues Format	40
1.11.12 OpcUaClient – Optimierung beim Erzeugen von Datentyp-Instanzen	40
1.11.13 OpcUaClient: Unterstützung von Grafik-Datentypen	40
1.11.14 OpcUaClient – Access.Advanced: Erweiterung bei ExtensionObjects	40
1.11.15 OpcUaClient – Access.Advanced: Erweiterung für Xml-Elemente	40
1.11.16 OpcUaClient – Access.Advanced: Korrektur für LocalizedTexts	40
1.11.17 OpcUaClient – Access.Direct: Korrektur für Strukturen	40
1.11.18 OpcUaClient – Access.Advanced: Optimierung für Strukturelemente	40
1.11.19 OpcUaClient – Access.Advanced: Unterstützung für EnumValueType	40
1.11.20 OpcUaClient – Access.Direct: Erweiterung um indizierten Zugriff	41
1.11.21 OpcUaClient – Access: Erweiterung um Attribut-Zugriff	41
1.11.22 OpcUaClient - Subscription: Korrektur beim Hinzufügen von Event-Items	41
1.11.23 OpcUaClient - Subscription: Neue Funktion Archive-Mode	41
1.11.24 OpcUaClient – MethodCall: Optimierung bei Ausgabe	41
1.11.25 OpcUaClient – MethodCall: Korrektur bei dynamischen Arrays	41
1.11.26 OpcUaClient – MethodCall: Korrektur für B&R-Autorisierungs-Methoden	41
1.11.27 OpcUaClient – MethodCall: Erweiterung für Xml-Elemente	41
1.11.28 OpcUaClient – MethodCall: Erweiterung für Datentp ‚TimeZoneDataType‘	41
1.12 RnCommTest V5.03	42
1.12.1 Korrektur Zertifikats-Anzeige	42
1.12.2 OpcUaClient: Optimierung der Anzeige und Eingabe von DateTime	42
1.12.3 OpcUaClient – Server-Properties: Änderung des Benutzers	42
1.12.4 OpcUaClient - Browse: Korrektur des Auswahl-Verhaltens	42
1.12.5 OpcUaClient - Attributes: Erweiterung bei Value-Anzeige	42

1.12.6 OpcUaClient - Access: Erweiterung bei ExtensionObjects	42
1.12.7 OpcUaClient - Subscription: Optimierung bei Event-Notification.....	42
1.12.8 OpcUaClient - Performance: Erweiterung um Action-Modus	42
1.12.9 OpcUaServer: Optimierung der Anzeige und Eingabe von DateTime	43
1.12.10 OpcUaServer – Settings: Neue Timer.....	43
1.12.11 OpcUaServer - AddressSpace: Erweiterung um Struktur-Arrays.....	43
1.12.12 OpcUaServer - AddressSpace: Erweiterung um Methoden.....	43
1.12.13 OpcUaServer - AddressSpace: Schnelles Setzen von Knoten-Parametern	43
1.12.14 OpcUaServer - AddressSpace: Korrektur beim Setzen von Source-Settings	43
1.12.15 OpcUaServer – Events: Mapping von Feld-Werten	43
1.12.16 OpcUaServer – Events: Zyklisches Feuern	43
1.12.17 OpcUaServer – Diag: Überarbeitung der Ausgaben bei MonitoredItems.....	43
1.12.18 OpcUaServer – Beispiel-Konfiguration: Neue Funktionen beispielhaft integriert	43
1.13 RnCommTest V5.02	44
1.13.1 OpcUa – Stack.....	44
1.13.2 OpcUaClient: Status auch als Hex-Zahl.....	44
1.13.3 OpcUaClient – Attributes: Korrektur beim Ermitteln der Browse-Pfade.....	44
1.13.4 OpcUaClient - Access: Erweiterung bei Enumerationen	44
1.13.5 OpcUaClient - Subscription: Erweiterung.....	44
1.13.6 OpcUaClient - Subscription: Erweiterung bei Event-Notification	44
1.13.7 OpcUaClient - Subscription: Änderung/Erweiterung bei EventFilter	44
1.13.8 OpcUaClient – Subscription: Korrektur bei Arrays	44
1.13.9 OpcUaClient – Access.Advanced: Erweiterung bei Enum	44
1.13.10 OpcUaClient – Access.Advanced: Korrektur bei Arrays.....	45
1.13.11 OpcUaClient – Access.Advanced: Korrektur bei ByteString-Wert.....	45
1.13.12 OpcUa-Client – MethodCall: Korrektur bei ByteString-Argument.....	45
1.13.13 OpcUa-Client – MethodCall: Korrektur bei ExtensionObject-Array	45
1.13.14 OpcUaClient – Access.Advanced: Importieren einer Datei	45
1.13.15 OpcUaClient – Access.Advanced: Exportieren als Datei	45
1.13.16 OpcUaServer - AddressSpace: Änderung des Layouts	45
1.13.17 OpcUaServer - AddressSpace: Hervorhebung von benutzerdefinierten Knoten.....	45
1.13.18 OpcUaServer - AddressSpace: Erweiterung um Event-Typen.....	45
1.13.19 OpcUaServer: Neue Seite „Events“	45
1.14 RnCommTest V5.01	46
1.14.1 Tool: Bessere Strukturierung durch Unterordner für Daten-Dateien	46
1.14.2 Tool: Korrektur beim Laden-/Speichern-Dialog	46
1.14.3 OpcUaClient: Neue Einstellungen zur Auflösung von Zeitangaben	46
1.14.4 OpcUaClient – Attributes: Erweiterung zum Ermitteln von Browse-Pfaden.....	46
1.14.5 OpcUaClient – Access: Korrektur bei Strukturen mit Int64, UInt64 und LocalizedText	46
1.14.6 OpcUaClient – New Endpoint: Korrektur beim Laden	46
1.14.7 OpcUaClient – OpcUa Monitor: Korrektur.....	46
1.14.8 OpcUaServer – Diag: Korrektur	47
1.14.9 OpcUaServer – Diag: Korrektur	47
1.14.10 OpcUaServer: Neue Einstellung zur Auflösung von Zeitangaben.....	47
1.14.11 OpcUaServer – AdressSpace - Source: Korrektur des Operanden für Datentyp ‚DateTime‘	47
1.14.12 OpcUaServer – AdressSpace - Source: Neue Action für Datentype ‚DateTime‘.....	47
1.14.13 OpcUaServer – AdressSpace: Unterstützung von Strukturen.....	47
1.14.14 OpcUaServer – AdressSpace: Änderbare Reihenfolge der Knoten	47
1.14.15 OpcUaServer – AdressSpace: Umwandlung von Object-Knoten in Struktur-Knoten und umgekehrt	48
1.14.16 OpcUaServer: Angabe der unterstützten Sprachen.....	48
1.14.17 ModBusTcpMaster: Erweiterung um Double.....	48
1.14.18 ModBusTcpMaster: Erweiterung um Swapping	48
1.14.19 ModBusTcpSlave: Erweiterung um Interpretation und Swapping	48
1.14.20 Pvi – Umstieg auf neue Version.....	48
1.14.21 Pvi – Erweiterung um DeviceType ‚ANSLTcp‘	48
1.14.22 Pvi – Korrektur: DeviceType ‚AR000‘ entfernt.....	48
1.14.23 Pvi – Logger: Korrektur	48
1.15 RnCommTest V5.00	49
1.15.1 Dialog „ShowText“: Erweiterung um Json-Format	49
1.15.2 Umstellung des OpcUa-Stacks	49
1.15.3 Neues Modul OpcUaServer	49
1.15.4 Neues Kapitel ‚Translations‘.....	49
1.15.5 OpcUa – Änderung der Ordner-Struktur und der Datei-Endungen	49
1.15.6 OpcUaClient – Verbindung: Korrektur.....	49
1.15.7 OpcUaClient – Browse: Optimierung	49
1.15.8 OpcUaClient – Attributes: Erweiterung.....	49
1.15.9 OpcUaClient – Access.Advanced: Korrektur.....	50
1.15.10 OpcUaClient – Access.Advanced: Korrektur.....	50
1.15.11 OpcUaClient – Access.Advanced: Optimierung.....	50
1.15.12 OpcUaClient – Access.Advanced: Optimierung.....	50

1.15.13 OpcUaClient – Subscription: Korrektur	50
1.15.14 OpcUaClient – Subscription: Korrektur	50
1.15.15 OpcUaClient – Subscription: Optimierung.....	50
1.15.16 OpcUaClient – Subscription: Korrektur	50
1.15.17 OpcUaClient – ServerDiag: Erweiterung.....	50
1.16 RnCommTest V4.07	51
1.16.1 Angabe einer Entwicklungs-Version.....	51
1.16.2 Korrektur der Wert-Eingabe	51
1.16.3 Dialog „ShowText“: Optimierung und Erweiterung	51
1.16.4 Neues Dialogfeld zum Verwalten von Listen	51
1.16.5 OpcUaClient – Endpoint-New: Speichern und Laden der Einstellungen.....	51
1.16.6 OpcUa-Client – Browsing	51
1.16.7 OpcUa-Client – Attribut-Liste: Erweiterung „Show as Text“	51
1.16.8 OpcUaClient – Access.Advanced: Optimierung bei der Anzeige von Arrays	51
1.16.9 OpcUaClient – Access.Direct: Auswahl der Nodetids	51
1.16.10 OpcUaClient – HistoricalAccess: Korrektur.....	51
1.16.11 OpcUaClient – ServerDiag: Erweiterung.....	51
1.17 RnCommTest V4.06	52
1.17.1 Hinweis auf die Hilfe.....	52
1.17.2 Korrektur der Element-Größen.....	52
1.17.3 Ethernet – Korrektur wegen IpV6.....	52
1.17.4 OpcUa-Client – Korrektur	52
1.17.5 OpcUaClient – Endpoint-New: Korrektur bei der Angabe der EndpointUrl.....	52
1.17.6 OpcUaClient – Endpoint-New: Angabe des Session-Timeouts.....	52
1.17.7 OpcUaClient – Endpoint-Discover: Anzeige des Server-Zertifikats.....	52
1.17.8 OpcUaClient – Endpoint-Search: Korrektur wegen IpV6.....	52
1.17.9 OpcUa-Client – Neuer Reiter „ServerDiag“	52
1.18 RnCommTest V4.05	53
1.18.1 Hilfe um den Anhang „ASCII-Codes“ erweitert.....	53
1.18.2 Ethernet – Erweiterung: Anzeige des Adapter-Titels.....	53
1.18.3 Ethernet – Erweiterung: Suche im Netzwerk.....	53
1.18.4 Pvi – Korrektur Browse-Fenster	53
1.18.5 OpcUaClient – Endpoint: Auswahl der Endpoint-URL.....	53
1.18.6 OpcUaClient – Endpoint: Angabe der bevorzugten Sprachen	53
1.18.7 OpcUaClient – Endpoint: Angabe einer Session-Name-Erweiterung.....	53
1.18.8 OpcUaClient – New Endpoint: Timeout für „Discover“	53
1.18.9 OpcUaClient – New Endpoint: Mehr Discover-Angaben	53
1.18.10 OpcUaClient – New Endpoint: Neuer Reiter „Search“	53
1.18.11 OpcUaClient – Zertifikats-Prüfung bei Connect	53
1.18.12 OpcUaClient – DirectAccess.Advanced: Korrektur	53
1.18.13 OpcUaClient – DirectAccess.Advanced: Anzeige des CommonCode	53
1.18.14 OpcUaClient – DirectAccess.Advanced: Anzeige eines Argument-Datentyps.....	53
1.18.15 OpcUaClient – DirectAccess: Neuer Reiter „Translate“	54
1.18.16 OpcUaClient – Attributes: Ermitteln der absoluten Browse-Pfade	54
1.18.17 OpcUa-Client – Subscription: Detailliertere Ausgabe bei Var-Items	54
1.18.18 OpcUa-Client – Subscription: Untergrenze Intervalle.....	54
1.18.19 OpcUa-Client – Subscription: Neue Einstellung für Items	54
1.18.20 OpcUa-Client – Subscription: Triggerung von Items	54
1.18.21 OpcUaClient – Certificates: Größere Schlüssel-Länge möglich.....	54
1.18.22 OpcUa-Client – Certificates: Angabe des Signatur-Algorithmus	54
1.18.23 OpcUa-Client – Hinweise: Umrechnung von UnitId zu CommonCode.....	54
1.19 RnCommTest V4.04	55
1.19.1 OpcUaClient – Discover.....	55
1.19.2 OpcUaClient – Erweiterte Einstellungen	55
1.19.3 OpcUaClient – Connection: Neue Einstellung.....	55
1.19.4 OpcUaClient – DirectAccess.Advanced: Korrektur	55
1.19.5 OpcUaClient – DirectAccess.Advanced: Speichern und Laden von Werten.....	55
1.19.6 OpcUaClient: Abfangen von NULL-Werten	55
1.19.7 OpcUa-Client – DirectAccess.Advanced: Anzeige von irregulären Enumerationen	55
1.19.8 OpcUa-Client – Subscription: Korrektur	55
1.19.9 OpcUa-Client – MethodCall: Anzeige des Pfades.....	55
1.19.10 OpcUa-Client – MethodCall: Enumeration als Argument	55
1.19.11 OpcUaClient – MethodCall: Unterstützung von komplexen Argumenten	55
1.19.12 OpcUaClient – MethodCall: Speichern und Laden von Einzel-Argumenten.....	55
1.19.13 OpcUaClient – MethodCall: Speichern und Laden von Argument-Listen.....	55
1.19.14 OpcUaClient – MethodCall: Behandlung von generischen Arrays	55
1.19.15 OpcUa-Client – Stack-Korrektur.....	56
1.20 RnCommTest V4.03	57
1.20.1 OpcUaClient – Attributes: Anzeige von lokalisierten Texten	57

1.20.2 OpcUa-Client – DirectAccess.Advanced: Anzeige eines Fortschrittsbalkens	57
1.20.3 OpcUa-Client – DirectAccess.Advanced: Anzeige von Byte() und ByteString	57
1.20.4 OpcUa-Client – DirectAccess.Advanced: Dialog zum Anzeigen von Texten	57
1.20.5 OpcUa-Client – Subscription: Korrektur	57
1.20.6 OpcUa-Client – Subscription: Korrektur	57
1.20.7 OpcUa-Client – MethodCall: Korrektur	57
1.21 RnCommTest V4.02	58
1.21.1 OpcUa-Client – Behandlung von ExtensionObjects (z.B. Strukturen)	58
1.21.2 OpcUaClient – Browse: Änderung des Browsings	58
1.21.3 OpcUaClient – Attributes: WriteMask als Text	58
1.21.4 OpcUaClient – Attributes: Neuer Menüpunkt „Go to Node“	58
1.21.5 OpcUa-Client – Access: Dialog für Texte	58
1.21.6 OpcUa-Client – DirectAccess.Advanced: Kopieren in die Zwischenablage	58
1.21.7 OpcUa-Client – Neuer Reiter „References“	58
1.21.8 OpcUa-Client – Subscription: Korrigieren der Settings durch den Server	58
1.21.9 OpcUa-Client – MethodCall: Anzeige und Editieren von generischen Arrays	58
1.21.10 OpcUa-Client – MethodCall: Anzeige von komplexen Output-Argumenten als Byte-Array	58
1.21.11 OpcUa-Client – Neuer Reiter „Performance“	58
1.21.12 OpcUa-Client – Certificates: Anzeige von erweiterten Feldern	58
1.22 RnCommTest V4.01	59
1.22.1 OpcUa-Client – Kopieren in die Zwischenablage	59
1.22.2 OpcUa-Client – Erweiterung des Attribut-Zugriffs	59
1.22.3 OpcUa-Client – Subscription	59
1.22.4 OpcUa-Client - Neue Endpoint-Einstellung	59
1.22.5 OpcUa-Client - Neue Client-Einstellung	59
1.22.6 OpcUa-Client - Neue Find-Einstellung	59
1.22.7 OpcUa-Client – Stack-Änderung	59
1.22.8 OpcUa-Client – Stack-Änderung	59
1.22.9 OpcUa-Client - Verbindung	59
1.22.10 OpcUaClient – Verbindung	59
1.23 RnCommTest V4.00	60
1.23.1 Umstellung der Entwicklungsumgebung	60
1.23.2 Umstellung des Frameworks	60
1.23.3 Umstellung des OpcUa-Stacks	60
1.23.4 OpcUa-Client - Neue Endpoint-Einstellung	60
1.23.5 OpcUa-Client – Änderung und Erweiterung für Events	60
1.23.6 OpcUa-Client – Änderung und Erweiterung für HistoricalAccess	60
1.23.7 OpcUa-Client - Monitor	60
1.24 RnCommTest V3.12	61
1.24.1 OpcUaClient – MethodCall	61
1.24.2 OpcUaClient – ServerInfo	61
1.24.3 OpcUaClient – Zertifikate	61
1.25 RnCommTest V3.11	62
1.25.1 OpcUaClient – Verbindung	62
1.25.2 OpcUaClient – Korrektur	62
1.25.3 OpcUaClient – MethodCall	62
1.26 RnCommTest V3.10	63
1.26.1 OpcUaClient – Verbindung	63
1.26.2 OpcUaClient – Discovery	63
1.26.3 ModbusMaster – Limitierung der Wert-Eingabe	63
1.26.4 Pvi – Default-Aktivierung	63
1.27 RnCommTest V3.09	64
1.27.1 Allgemein – Fenster-Größe	64
1.27.2 OpcUaClient – Attribute-Anzeige	64
1.27.3 OpcDaClient – Reiter wird versteckt	64
1.28 RnCommTest V3.08	65
1.28.1 OpcUaClient – Discovery	65
1.28.2 OpcUaClient – Attribute-Anzeige	65
1.28.3 OpcUaClient – Subscription	65
1.28.4 OpcUaClient – Subscription - ItemSettings	65

1 Revisionsgeschichte

Die Revisionsgeschichte ist ab V3.08 enthalten.
Ab V7.00 ist sie aus der Hilfe in diese Datei extrahiert worden.

1.1 RnCommTest V8.00

1.1.1 Allgemein

1.1.1.1 Umstellung des Dot.Net-Frameworks auf 4.8

Das Tool wurde von Dot.Net-Framework 4.7.2 auf 4.8 umgestellt. Siehe [Systemanforderungen](#) und [Versionen verwendeter Software](#).

1.1.1.2 Neue Haupt-Tab-Leiste

Da durch die immer mehr werdenden Module die obere Tab-Leiste unübersichtlich wurde, gibt es jetzt eine Haupt-Tab-Leiste, die die Module nach Protokollen gruppiert. Ausserdem wird jetzt durch eine andere Darstellungsart die selektierte Tab-Seite besser hervorgehoben.

1.1.2 OpcUa

1.1.2.1 Umstellung des OpcUa-Stacks auf 1.05.374.54

Es wurde von OpcUa-Stack 1.04.368.53 auf 1.05.374.54 umgestellt. Dieser setzt die Funktionalitäten der Spezifikation V1.05 teilweise schon um. Noch nicht implementierte Funktionalitäten von V1.05 werden zukünftig schrittweise umgesetzt.

Nach Vorgabe der Opc-Foundation muss V1.05 kompatibel zu V1.04 sein, so sollte auch die Funktionalität dieses Tools kompatibel zur vorigen Version sein.

Siehe [Versionen verwendeter Software](#).

1.1.2.2 Einheitliches Verzeichnis für alle PubSub-Konfigurationen

Alle PubSub-Konfigurationen aus Client, Subscriber und Server sind jetzt nicht mehr in Unterverzeichnissen der einzelnen Module abgelegt, sondern nur noch im Verzeichnis

RnCommTest\OpcUa\PubSubCfg. Da oft eine Konfiguration in einem Modul gespeichert und in einem anderen Modul geladen werden muss, ist dies jetzt ohne ständigen Verzeichniswechsel in den Speichern/Öffnen-Dialogen möglich.

1.1.2.3 OpcUaPubSubConfigurator

1.1.2.3.1 Erweiterung des Konfigurations-Typen

In der Spezifikation V1.05 wurde ein erweiterter Datentyp für PubSub-Konfigurationen definiert, welcher in den vorherigen Versionen dieses Tools nicht unterstützt wurde (es gab eine Exception beim Laden/Importieren).

Aufgrund der Umstellung auf OpcUa-Stack 1.05.374.54 können nun auch diese geladen/importiert werden. Intern wird aber immer noch mit dem ersten Datentyp gearbeitet/abgespeichert.

1.1.2.3.2 Erweiterung um Parameter ‚SecurityMode‘

In einer WriterGroup gibt es den Parameter *SecurityMode*, welcher bisher nicht angezeigt wurde. Jetzt ist dieser sichtbar und kann auch editiert werden. Allerdings ist die Sicherheit durch Signierung und Verschlüsselung in den meisten PubSub-Implementierungen (wie auch in diesem Tool) noch nicht verfügbar. Deshalb macht eine Einstellung ausser *Invalid* oder *None* keinen Sinn (zur eingeschalteten Sicherheit würden noch weitere Parameter gehören, die aber nicht angezeigt werden).

Der Parameter wird lediglich angezeigt, um die Einstellung korrigieren zu können, wenn sie aus welchem Grund auch immer nicht auf *Invalid* oder *None* steht.

Siehe OpcUaClient/PubSubConfigurator/Konfiguration/WriterGroup/[SecurityMode](#).

1.1.2.3.3 Erweiterung um Datentyp UInt64 für PublisherId laut V1.05

Laut Spezifikation V1.04 des Header-Layouts UADP-Periodic-Fixed **musste** der *PublisherIdType* sowohl in der *Connection* (für Publisher) als auch im *Reader* (für Subscriber) als *UInt16* ausgewiesen sein. Ab der Spezifikation V1.05 darf der *PublisherIdType* auch als *UInt64* gesetzt sein. Hintergrund: Damit kann man besser eine eindeutige *PublisherId* setzen, etwa indem man die MAC-Adresse (6 Byte) des

Geräts in den Wert codiert. Siehe WriterGroup/[HeaderLayoutUri](#), Reader/[HeaderLayoutUri](#) und Kontextmenü der Connections/[Set HeaderLayout](#).

1.1.2.3.4 Datentyp-Prüfung bei DragAndDrop optimiert

Wenn ein Source- oder Target-Node eines Fields der Konfiguration per DragAndDrop aus dem Client-Browse-Baum herübergezogen wird, wird erst geprüft, ob der Datentyp passt. Für den Datentyp eines Fields gibt es die Attribute `Datatype` (als `NodeId`) und `BuiltInType` (als Enumeration). Es wurde erst der `Datatype` geprüft und wenn dieser `Null` war dann `BuiltInType`.

Leider gibt es aus Fremdsystemen importierte Konfigurationen, bei denen zwar der `BuiltInType` korrekt besetzt ist, der `Datatype` aber nicht `Null`, sondern als `ns=0;i=0` gesetzt ist.

Bis jetzt war in diesem Fall ein DragAndDrop nicht möglich. Jetzt wird auch diese Möglichkeit geprüft und zugelassen, wenn der Datentyp zusammenpasst.

1.1.2.4 OpcUaClient

1.1.2.4.1 Korrektur wegen AR6.00

Beim Verbinden zum Server eines B&R-AR6.00 oder höher wurde ein Fehlverhalten des Clients festgestellt, wegen dem keine Verbindung aufgebaut werden konnte.

Details: Schon während des Verbindungs-Aufbaus, aber noch vor dem Aktivieren der Session wurde der Server-Status gelesen. Diese zeitabhängige Funktion führte meistens zu einer Fehlermeldung (z.B. *SessionNotActivated*) und der sofortigen Schliessung der Session seitens des Clients, so daß die Verbindung nicht zustande kam. Ob dieses Fehlverhalten am neuen Client-Stack oder am AR-Server lag, konnte nicht eindeutig festgestellt werden.

Jetzt wird der Server-Status erst nach der Session-Aktivierung gelesen, so dass die Verbindung aufgebaut werden kann.

1.1.2.4.2 Verändern der Benutzer-Reihenfolge in der Anmelde-Liste

Bei der Liste der Anmelde-Benutzer kann jetzt die Reihenfolge der Einträge über das Kontextmenü verändert werden. Siehe [New Endpoint/Authentication/User](#).

1.1.2.4.3 Korrekturen beim Akzeptieren von Server-Zertifikaten

War die Option [Accept Server Certificate Automatically](#) nicht aktiviert, konnte es seit den letzten Versionen vorkommen, dass das zugehörige Dialogfeld zum manuellen Akzeptieren zweimal hintereinander angezeigt wurde. Auch wenn beide Male auf `Accept` geklickt wurde, wurde die Verbindung schon nach dem ersten Mal abgelehnt.

Dies ist jetzt korrigiert: Das Dialogfeld erscheint nur einmal und bei Klick auf `Accept` wird die Verbindung akzeptiert.

Außerdem wird ein Server-Zertifikat bei der Policy `None-None` jetzt immer automatisch akzeptiert.

1.1.2.4.4 Neue Such-Funktion FindNodeId

Im Kontext-Menü des Browse-Baums gibt es sich jetzt den neuen Punkt [Find NodeId](#).

Im Gegensatz zur schon bestehenden Funktion [Find](#), welche tatsächlich Knoten durchsucht, wird bei der neuen Funktion von dem Knoten weg der aufwärtsgerichtete Pfad ermittelt. Dies ist wesentlich schneller. In diesem Zuge wurde auch die Speicherung der Suchtexte beider Funktionen optimiert (das Zeichen `|` zur Trennung der einzelnen Suchtexte (welches auch bei `NodeId`'s verwendet wird) wurde durch `|` ersetzt, so dass die einmal verwendeten Texte beim nächsten Start des Tools auch wieder richtig zur Verfügung stehen.

Außerdem wird diese Funktion jetzt auch bei [Go to Node](#) in der Attribut-Liste verwendet.

1.1.2.4.5 Optimierung bei Browse-Icons

Bei Methoden-Knoten kann jetzt schon anhand des Icons im Browse-Baum gesehen werden, ob die Methode vom aktuellen Benutzer aufgerufen werden darf (wird über das Attribut `UserExecutable` ermittelt). Siehe [Browse](#).

1.1.2.4.6 Optimierung/Erweiterung bei der Anzeige/Editierung von Attributen

Die Bit-Texte von bitcodierten Attributen (z.B. `WriteMask`, `AccessLevel` usw.) sind nicht mehr fest hinterlegt, sondern werden jetzt einmalig pro Session vom Server gelesen. Somit werden auch zukünftige Erweiterungen (so wie bei der aktuellen Spec V1.05 geschehen) von Bits korrekt dargestellt.

Bei einem Doppelklick auf ein Attribut in der Attribut-Liste wurde bis jetzt ein Dialogfeld zur detaillierten Anzeige des Attribut-Werts geöffnet. Wenn das Attribut zum Schreiben freigegeben ist (hinterlegt im Attri-

but `UserWriteMask`), musste zum Schreiben der Reiter `Attributes` (siehe [Access/Attributes](#)) verwendet werden. Nun ist zusätzlich auch eine bequemere Editierung im Dialogfeld möglich. Siehe [Attribut-Liste](#).

Die Änderung von bitcodierten Attributen muss nun nicht mehr als Dezimalzahl erfolgen, sondern wird als boolsche Liste mit Bit-Nummer, Bit-Wert, Bit-Text und Gesamt-Wert dargestellt. So werden Fehleingaben vermieden. Siehe [Attribut-Liste/Show in dialog](#).

Die `RolePermissions`-Attribute (`RolePermissions` und `UserRolePermissions`, Arrays von `RolePermissionType`) eines Knotens, welche Server It. Spezifikation V1.05 optional unterstützen können und die die OpcUa-eigene Berechtigungs-Vergabe darstellen (siehe Spec Part 3), können jetzt auch mit einem Doppelklick in einem eigenen Dialogfeld geöffnet werden. Diese Werte werden im Dialogfeld nun nicht mehr nur in ihrer natürlichen Form, sondern auch aufgeschlüsselt als Klartext dargestellt, was die Interpretation sehr vereinfacht. Zusätzlich können sie je nach Recht auch geändert werden. Siehe [Zugriff auf RolePermissions](#).

Kann ein Attribute nicht korrekt gelesen werden, wird diese Zeile normalerweise rot hinterlegt. Da das Attribute `RolePermissions` von den meisten Rollen aus Sicherheits-Aspekten nicht gelesen werden darf (nur von der Rolle `SecurityAdmin`), wird es nicht rot hinterlegt, um Verwirrung vorzubeugen und die Liste überschaulich zu halten.

1.1.2.4.7 Erweiterung beim Auflösen von Datentypen

Von der Foundation spezifizierte Datentypen, die dem Stack implizit bekannt sein müssen (Namespace `Opc.Ua`) und dessen Baupläne deshalb nicht vom Server veröffentlicht werden, werden nun generisch aufgelöst. Damit werden auch zukünftige Datentypen behandelt, wenn sie neu hinzukommen.

1.1.2.4.8 Optimierung bei Event-Filter-Anzeige

Bei der Anzeige des `SelectClause`-Filters eines Event-Items werden jetzt nur noch Knoten vom Typ `ObjectType` und `Variable` angezeigt. Andere Typen (z.B. `Methods`) sind hier nicht benötigt, weil sie nicht hinzugefügt werden können. Siehe [Event Filter \(SelectClause\)](#).

1.1.2.4.9 Optimierung bei ServerDiag

Stellte der verbundene Server den Knoten `ServerDiagnosticsSummary` (`i=2275`) nicht zur Verfügung, wurde beim Start der Server-Diagnose eine Exception geworfen. Diese wird jetzt abgefangen und eine entsprechende Ausgabe gemacht.

1.1.2.5 OpcUaServer

1.1.2.5.1 Anpassungen für neuen Stack wegen RolePermissions

Der neue Stack unterstützt teilweise schon die neuen Funktionalitäten der `RolePermissions` It. Spezifikation 1.05 (OpcUa-eigene Rechte-Vergabe, siehe Spec Part 3). Diese sollen beim Tool aber keine Anwendung finden, da hier nur eine sehr einfache Benutzer-Verwaltung implementiert ist (siehe [UserManagement](#)). Aus diesem Grund wurden mehrere Änderungen nötig. Der Server sollte dadurch kompatibel zur vorherigen Version sein.

1.1.2.5.2 Korrekturen beim Laden von Fremdmodellen

Bei manchen Fremdmodellen (insbesondere aus AS6.00) traten Exceptions auf. Dies wurde behoben. Details: Aufgrund von AS-Projekt-Einstellungen werden manche Knoten erst zur Laufzeit des AR erzeugt, die dann aber unbedingt benötigt werden. Dieser Umstand musste berücksichtigt werden.

1.1.2.6 OpcUaBrMapper

1.1.2.6.1 Erweiterung um das uad-Format von AS6.00

Ab AS6.00 hat sich das Format der `DefaultView`-Datei (*.uad) geändert. Dieses Format konnte mit vorigen Versionen des Tools nicht eingelesen werden (die Tabelle der Knoten blieb leer). Jetzt wird zusätzlich das neue Format erkannt und eingelesen. Siehe [OpcUaBrMapper/DefaultView](#). Außerdem müssen für dieses neue Format die zusätzlichen Namespaces manuell eingelesen werden. Siehe [OpcUaBrMapper/Load additional namespaces](#).

1.1.2.7 Neues Modul OpcUaDevConfigurator

Mit dem neuen Modul [OpcUaDevConfigurator](#) kann bequem ein Fremdgerät (z.B. die B&R-Geräte `Switch 0ACST052.1` oder `Buskoppler X20BC008T`) konfiguriert werden, welches dazu einen

OpcUa-Server bietet. Der Modul-Client verbindet sich dabei auf diesen Server und legt Benutzer an, setzt Parameter, ruft Methoden auf usw. Die nötigen Aktionen können vom Anwender dabei frei parameterisiert werden.

1.2 RnCommTest V7.06

1.2.1 Allgemein

1.2.1.1 Neu: Versions-Informationen verwendeter Software

Es wurde der neue Absatz [Versionen verwendeter Software](#) im Kapitel Allgemeines aufgenommen.

1.2.1.2 Neues Modul „TextFormatter“

Maschinell erstellte Dateien im Xml- oder Jason-Format beinhalten oft keine Zeilenumbrüche und Einrückungen, was sie für einen Menschen unleserlich machen. In diesem Modul können beliebig viele [Text-Fenster](#) geöffnet werden, welches eine Formatierung in eine lesbare Form unterstützt. Siehe [TextFormatter](#).

1.2.1.3 Text-Fenster

Der [Text-Fenster](#)-Dialog wurde mit dem Button ‚Paste text from Clipboard‘ erweitert. Damit kann ein in der Zwischenablage befindlicher Text in das Dialogfeld übernommen werden. Außerdem wurde die XML-Formatierung überarbeitet, so dass sie jetzt besser funktioniert.

1.2.2 OpcUa

1.2.2.1 Korrektur am OpcUa-Stack

Im Stack V1.04.368.53, der seit RnCommTest V7.03 eingesetzt wird, wurde ein Fehlverhalten festgestellt. Dieses führte u.a. dazu, dass am Server PropertyNodes ‚EngineeringUnits‘ und ‚EURange‘ nicht mehr korrekt eingefügt werden konnten. Da es sich dabei um eine oft verwendete Hilfs-Funktion handelt, können auch andere Einflüsse nicht ausgeschlossen werden. Dieses Verhalten wurde im Stack korrigiert.

1.2.2.2 OpcUaClient

1.2.2.2.1 Optimierung bei der Sortierung von Sub-Knoten

Bei der Sortierung von Knoten im Browser wurde die Groß-/Kleinschreibung berücksichtigt, so dass Namen mit „a“ nach „Z“ einsortiert wurden. Dies wurde geändert, so dass Namen mit „a“ nun nach „A“ einsortiert werden. So können Knoten schneller gefunden werden. Ebenso werden Sub-Knoten eines Arrays jetzt richtig nach aufsteigendem Index sortiert (bisher war die Reihenfolge textbasiert: 0,10..19, 1, 20..29, 2 usw).

1.2.2.2.2 Korrektur beim Suchen

Beim Suchen eines Knotens im Browse-Fenster ([Find...](#)) wurden manche Unterknoten nicht durchsucht und dadurch ein tatsächlich vorhandener Knoten nicht gefunden.

Hintergrund: Um die Suche enorm effizienter zu machen und ein nochmaliges Durchsuchen von bereits durchsuchten Knoten zu verhindern (derselbe Knoten kann in mehreren Zweigen liegen), wurden sich schon durchsuchte Knoten gemerkt. Dies geschah bis jetzt nur durch den Nodeld-Identifizier. Da dieser in verschiedenen Namespaces vorkommen konnte, wurde Knoten mit demselben Identifizier, aber einem anderen Namespace zugehörig, nicht durchsucht.

Jetzt geschieht die Speicherung über die gesamte Nodeld, welche den Namespace-Index enthält. So werden alle Unterknoten durchsucht.

1.2.2.2.3 Erweiterung für WriteOnly-Variablen

Bisher konnten Werte von WriteOnly-Variablen nur über den Reiter `Standard` beschrieben werden. Im Reiter `Advanced` wurde kein Wert angezeigt. So war es z.B. unmöglich, WriteOnly-Strukturen zu beschreiben.

Jetzt wird der selektierte Variablen-Knoten mit einem selbst erzeugten Wert des Datentyps angezeigt (alle Sub-Werte werden mit 0 initialisiert, da die aktuellen Werte ja nicht gelesen werden können), welcher besetzt und geschrieben werden kann. Einmal gesetzte Werte werden gespeichert, so dass der zuletzt geschriebene Werte erhalten bleibt. Auch das Speichern über das Kontext-Menü ist so möglich. Siehe [WriteOnly-Variablen](#).

1.2.2.2.4 Neuer Menüpunkt im Browse-Fenster: Call method without arguments

Mit dem neuen Menüpunkt [Call method without arguments](#) kann eine Methode ohne Argumente schnell und bequem aufgerufen werden, ohne sie vorher in die [Methoden-Liste](#) aufzunehmen.

1.2.2.2.5 Neuer Menüpunkt im Browse-Fenster: Add all Methods

Mit dem neuen Menüpunkt [Add all Methods](#) können alle unter dem ausgewählten Objekt-Knoten liegenden Methoden auf einmal in die [Methoden-Liste](#) übernommen werden.

1.2.2.2.6 Optimierung beim Auflösen eines Datentyps

War ein Datentyp mit gleichem Namen in mehreren Datentyp-Dictionaries vorhanden, so wurde unter bestimmten Umständen die falsche oder keine Deklaration erkannt und es endete mit einer Fehlermeldung. Die Suche wurde dahingehend optimiert.

1.2.2.2.7 Korrektur im Reiter Attributes/Advanced

Beim Anwählen eines Knotens mit Datentyp `DateTime` wird die Anzahl der Ticks berechnet und in der Spalte `Description` ausgegeben. Wurde dann durch den Button Read der Wert erneut gelesen, wurde die Anzahl der Ticks nicht aktualisiert. Dies ist behoben.

1.2.2.2.8 Erweiterung im Reiter MethodCall

Hat ein Methoden-Argument einen von einem Standard-Datentyp abgeleiteten Datentypen, so wird es jetzt korrekt eingefügt. Ebenso kann so ein Argument jetzt mit einem Zufallswert besetzt werden.

1.2.2.2.9 Dialogfeld für Methoden-Argumente (Show in dialog)

Ist der Wert des ausgewählten Arguments kein einfacher Datentyp, so wird bei einem Doppelklick ein Dialogfeld zum Bearbeiten geöffnet, während einfache Datentypen direkt eingegeben werden können. Durch den neuen Kontext-Menüpunkt [Show in dialog](#) wird immer das Dialogfeld geöffnet, welches zusätzliche Funktionalitäten bietet. So kann z.B. ein String aus einer Datei importiert oder exportiert werden.

1.2.2.3 OpcUaSubscriber

1.2.2.3.1 Optimierung der Wert-Anzeige


Wenn viele Felder empfangen wurden, flackerte die Werte-Liste durch die Aktualisierung unter Umständen sehr stark, was die Analyse sehr erschwerte. Durch Verwendung eines anderen Anzeige-Controls wurde das Flackern eliminiert.

1.2.2.4 OpcUaServer

1.2.2.4.1 Änderung beim Standard-Wert für EngineeringUnit

Bis jetzt wurde beim Hinzufügen eines [Property EngineeringUnit](#) die UnitId 5066068 (Millimeter) vorgewählt. Da dies in einigen Fällen (z.B. beim Import von Fremdmodellen) zu Verwirrung führte, wird jetzt die UnitId 0 vorgewählt. Somit ist klar, dass die UnitId nicht vergeben wurde.

1.2.2.4.2 Object-Knoten mit EventNotifier

Bei einem Object-Knoten kann jetzt das Attribut `EventNotifier` auf `1=SubscribeToEvents` gesetzt werden, so dass er Events feuern und von einem Client auf Events abonniert werden kann. Dies ist auch am Icon sichtbar, welches dann einen Blitz enthält: 

Siehe [EventNotifier](#).

Die Beispiel-Konfiguration [Example_General](#) wurde darum erweitert (siehe [Externes Modell](#)).

1.2.2.4.3 Feuern von Events an eigenen und externen Object-Knoten

Bei einer Event-Instanz können bis zu 3 Object-Knoten angegeben werden, welche das Event zusätzlich zum Server-Knoten feuern. Siehe [Additional notifiers](#).

Die Beispiel-Konfiguration [Example_General](#) wurde darum erweitert (siehe [EventType](#)).

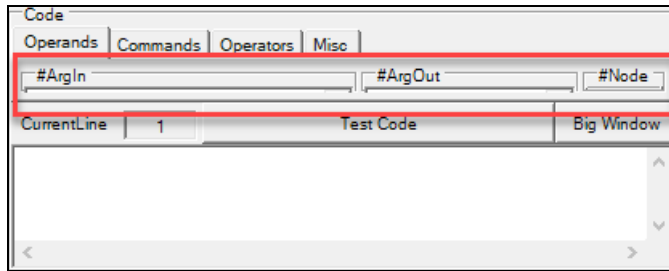
1.2.2.4.4 Erweiterung für Enumerations

Bisher konnte eine Enumerations-Variable nur dadurch erzeugt werden, dass an einem `Int32`-Knoten ein Property-Knoten `EnumStrings` hinzugefügt wurde. Damit waren nur reguläre Enumerationen möglich, die auch nur durch diesen Property-Knoten definiert waren. Siehe [Property EnumStrings](#).

Jetzt gibt es die Möglichkeit, den Datentyp auf einen vordefinierten Enumerations-Typ zu setzen. Siehe [Transform Variable to Enumeration](#).

1.2.2.4.5 Korrektur am Method-Code-Fenster

Ab V7.04 war der Reiter für Operands im Method-Code-Fenster verkleinert und so nicht optimal bedienbar:



Dies wurde behoben.

1.2.2.4.6 Neues Kommando bei Method-Code: Feuern eines Events

Durch das neue Code-Kommando #FireEvent in [Kommandos](#) kann eine Methode eine parametrisierte Event-Instanz feuern. Die Event-Instanz kann bequem über ein Auswahlfeld gewählt werden. Die Beispiel-Konfiguration [Example_General](#) wurde darum erweitert (siehe [Methoden](#)).

1.2.2.4.7 Neues Kommando bei Method-Code: Setzen des Rückgabe-Status

Durch das neue Code-Kommando #Status in [Kommandos](#) kann eine Methode einen definierten Status zurückgeben. Oft verwendete Stati können dabei bequem über ein Auswahlfeld gewählt werden. Die Beispiel-Konfiguration [Example_General](#) wurde darum erweitert (siehe [Methoden](#)).

1.2.2.4.8 Erweiterungen und Optimierungen für externe Modelle

Der Import und die Behandlung von externen Modellen wurde stark erweitert und optimiert (siehe [Extern models](#)).

Dies bietet viele Vorteile, z.B. können Knoten aus externen Modellen zum Teil wie selbst angelegte Knoten parametrisiert werden:

- Der AccessLevel kann nicht mehr nur für alle Benutzer gemeinsam, sondern für jeden Benutzer einzeln verändert werden
- Variablen-Werte können belebt werden (Init-Wert setzen, Runtime-Wert setzen und zyklisch ändern usw.)
- Methoden können mit Code versehen werden
- Events können an externen Object-Knoten gefeuert werden
- Diag-Ausgaben können aktiviert werden (Ausgaben bei Read, Write, Execute usw.)

Hinweis: Da die Parametrierung der externen Knoten gespeichert und geladen werden muss, haben sich die Zeiten dafür verlängert.

Wichtiger Hinweis: Beim Laden einer Konfiguration mit externen Modellen werden jetzt einige Parameter anders behandelt. Bei alten Konfigurationen aus früheren RnCommTest-Versionen ist es deshalb notwendig, alle externen Modelle zu entfernen und neu hinzuzufügen, um die korrekte Funktionalität zu gewährleisten. Details dazu siehe [Hinweise zum Hinzufügen/Laden einer Konfiguration](#).

Die Beispiel-Konfiguration [Example_General](#) wurde darum erweitert.

1.2.2.4.9 Fehlerbehebung bei BrowseNames

Die in einem externen Modell verwendeten Namespace-Indizes der Knoten beziehen sich immer auf die Namespaces des Modells. Beim Zusammenführen von mehreren Modellen erweitert sich die Namespace-Tabelle und deshalb verschieben sich auch manche Namespace-Indizes.

Beim Import müssen deshalb die Namespace-Indizes auf die gesamte Tabelle angepasst werden. Bei den Nodeld's wurde diese Anpassung während des Imports schon gemacht, bei den BrowseNames noch nicht. Dies wurde behoben.

1.2.2.5 OpcUaBrMapper

1.2.2.5.1 Erweiterung beim Einlesen eines Fremdmodells

Wenn es im Fremdmodell Knoten gab, deren Parent-Knoten in einem anderen Fremdmodell definiert ist, wurden sie nicht angezeigt. Sie konnten dann natürlich auch nicht gemappt werden. Jetzt wird in diesem Fall ein Dummy-Knoten als Parent eingefügt. So kann die Hierarchie trotzdem dargestellt werden und ein Mapping erfolgen. Siehe [Node-Liste](#).

1.2.2.5.2 Erweiterung beim Einlesen einer DefaultView für AS6.00

Bis AS4.12 waren alle verwendeten NamespaceUris aller Fremdmodelle in der *.uad-Datei hinterlegt. Ab AS6.00 ist dies nicht mehr der Fall. Sie sind dann in einer eigenen Config-Datei mit der Endung *.uadcfg enthalten. Da zum Erkennen bzw. Setzen eines Mappings aber die Namespaces des Fremdmodells benötigt werden, müssen diese nun zusätzlich geladen werden. Siehe [Load additional namespaces](#).

1.3 RnCommTest V7.05

1.3.1 OpcUa - PubSubConfigurator

1.3.1.1 Korrektur beim Import einer Datei mit mehreren Konfigurationen

Beim PubSub-Configurator, welcher in mehreren OpcUa-Modulen verwendet wird, wurde ein Import-Fehler behoben.

Eine uabinary-Datei kann mehrere Konfigurationen enthalten. Jede davon kann nur Publisher (Writer), nur Subscriber (Reader) oder auch beides enthalten. Je nach Modul (OpcUaClient, OpcUaServer oder OpcUaSubscriber) sollten bei der Import-Funktion nur bestimmte Konfigurationen importiert und die anderen ausgefiltert werden.

Wurde eine Konfiguration ausgefiltert, wurden die folgenden Konfigurationen der Datei nicht mehr geprüft und damit auch nicht mehr importiert. Dies ist jetzt behoben.

1.3.1.2 Optimierung für GroupVersion

Der Zeitstempel einer `GroupVersion` wird in der Spalte `Description` als Klartext angezeigt. Wurde dieser Parameter geändert, so wurde die Klartextanzeige nicht sofort aktualisiert. Dies ist jetzt behoben.

1.3.1.3 Anzeige der Properties

Manche Elemente einer PubSub-Konfiguration (z.B. Connection oder WriterGroup) besitzen optionale Properties als dynamische Liste, welche herstellerspezifische Parameter beinhalten können. Diese werden jetzt angezeigt und können auch editiert werden. Siehe [Editor für Konfigurations-Properties](#).

1.3.2 OpcUaClient

1.3.2.1 Optimierung bei Anzeige einer Enumeration

Ein Enumerations-Wert wird als `Int32` übertragen. Der logische Wertebereich ist durch die Enumeration auf deren Auflistung beschränkt. Nichtsdestotrotz könnte auch ein Wert außerhalb der Enumeration gesetzt sein. So könnte eine Enumeration, die mit Werten von 1..3 deklariert ist, auch mit 0 oder 4 besetzt sein (auch wenn dies logisch keinen Sinn macht).

Normalerweise wird bei diesem Tool der Wert einer Enumeration als Auswahlfeld mit allen gültigen Enumerations-Werten angezeigt (siehe Parameter [Interpret enumerations](#)). So kann der Benutzer nur gültige Werte setzen. Außerdem werden in der Spalte `Description` alle gültigen Werte mit deren Text angegeben:

Name	Value	Description
eRegular	0=nTEST_ENUM_BERLIN	0=nTEST_ENUM_BERLIN, 1=nTEST_ENUM_HAMBURG, 2=nTEST_ENUM_KOELN, 3=nTEST_ENUM_MUENCHEN, 4=nTEST_ENUM_REGENSBURG, 5=nTEST_ENUM_LANDSHUT
	0=nTEST_ENUM_BERLIN	
	1=nTEST_ENUM_HAMBURG	
	2=nTEST_ENUM_KOELN	
	3=nTEST_ENUM_MUENCHEN	
	4=nTEST_ENUM_REGENSBURG	
	5=nTEST_ENUM_LANDSHUT	

Ist der aktuelle Wert aber nicht in der Enumeration enthalten, so wird statt des Auswahlfelds immer ein `Int32`-Eingabefeld verwendet, um so den aktuellen Wert korrekt anzuzeigen. In der `Description` stand dabei bisher nur „`Int32 (Enum)`“. Um diesen Umstand etwas klarer darzustellen, werden jetzt in diesem Fall in der `Description` ebenfalls alle gültigen Werte mit deren Text angegeben:

Name	Value	Description
eRegular	123	Int32 (Enum: 0=nTEST_ENUM_BERLIN, 1=nTEST_ENUM_HAMBURG, 2=nTEST_ENUM_KOELN, 3=nTEST_ENUM_MUENCHEN, 4=nTEST_ENUM_REGENSBURG, 5=nTEST_ENUM_LANDSHUT)

So kann der Benutzer sehen, welche gültigen Werte es gibt und den Wert entsprechend setzen.

Siehe `AttributesAccess/Advanced/Enumerations`.

1.3.2.2 Erweiterung bei Wert-Editor

Bei einem dynamischen Array von den Datentypen `LocalizedText` und `QualifiedNames` kann jetzt auch die Element-Anzahl verändert werden.

1.3.2.3 Korrekturen und Erweiterungen für TemporaryFileType

Das Lesen ([Read to local file](#)) eines `TemporaryFileType` führte am Ende einen `Close` auf die temporär erzeugte Datei aus, obwohl laut Spezifikation eigentlich eigentlich ein `CloseAndCommit` am Datei-Knoten ausgeführt werden müsste. Manche Server hatten damit ein Problem. Dies ist jetzt korrigiert.

Hinweis: Beim Schreiben ([Write from local file](#)) war es schon korrekt implementiert.

Außerdem wurden beim Zugriff auf die Schreib-Methode der temporär erzeugten Datei immer eine von der Spezifikation vorgeschlagene MethodId verwendet. Manche Server vergeben aber eigene Id's, welche nun extra ausgelesen und verwendet werden.

Für File-Methoden des Typs `FileType` gibt es die Einstellung [Set file handle for file transfer methods automatically](#), welche den beim Open als Ausgangs-Argument zurückgegebenen `FileHandle` automatisch auf das entsprechende Eingangs-Argument anderer Methoden desselben Files (z.B. auf die `Close`-Methode) übernimmt.

Diese bequeme Funktionalität wurde jetzt auch für Methoden vom `TemporaryFileType` übernommen.

1.3.2.4 MethodCall: Erweiterungen und Optimierungen

Eine Methode kann jetzt in der Liste nach oben oder unten verschoben werden. Das ist besonders beim Arbeiten mit `Call list` hilfreich. Siehe [Methoden-Liste](#).

Wird eine Methode zur Liste hinzugefügt, so wird der Listen-Name automatisch nach dem Format `ObjectId.MethodId` zusammengestellt, um eine bessere Unterscheidbarkeit zu haben. Wenn nun (wie bei manchen Servern üblich, vor allem bei generisch erzeugten Methoden) die MethodId selbst schon mit der ObjectId beginnt, so war der bis jetzt erstellte Name ziemlich sperrig. In diesem Fall wird jetzt nur noch die MethodId herangezogen, was dann ebenfalls eindeutig ist, den Listen-Namen aber kürzer hält.

Hat ein Eingangs-Argument einer Methode den Datentyp `BaseDataType`, so kann jetzt optional ein Datentyp definiert werden (siehe [BaseDataType](#)).

1.3.3 OpcUaSubscriber

1.3.3.1 Anzeige einer Statistik aller empfangenen Werte

Aus der Statistik kann erkannt werden, viele Felder aus wievielen Datasets von wievielen Publishern empfangen wurden. Das erleichtert die schnelle Auswertung, ob alle erwarteten Daten empfangen wurden (siehe `OpcUaSubscriber/Values`).

1.3.3.2 Sortierung der empfangenen Werte

Durch Klick auf die Spalte `FieldName` kann die Liste der empfangenen Werte jetzt sortiert werden. Dadurch wird die Anzeige übersichtlicher und erleichtert die Suche bei vielen Werten (siehe `OpcUaSubscriber/Values`).

1.3.4 OpcUaServer

1.3.4.1 Optimierungen für FileType

Beim Stoppen des Servers werden jetzt alle offenen Dateien geschlossen.

Bei einem Knoten des Typs `FileType` (siehe [File](#)) werden jetzt die Zugriffsrechte für dessen unterlagerten Methoden und Variablen korrekt gesetzt. Bei der Überprüfung der Rechte zum Lesen/Schreiben des Knotens wird jetzt auch korrekt der `BadUserAccessDenied`-Status zurückgegeben.

1.3.4.2 AddressSpace: Erweiterung um TemporaryFileTransferType

Zusätzlich zum `FileType` kann im Baum jetzt auch ein Knoten vom Typ `TemporaryTransferFileType` hinzugefügt werden. Damit kann der lesende oder schreibende Zugriff auf eine temporäre Datei gewährt werden. Siehe [Temporary File](#).

Die Beispiel-Konfiguration [Example_General](#) wurde um solch einen Knoten erweitert.

1.3.5 OpcUaBrMapper

1.3.5.1 Korrekturen beim Einlesen des Fremdmodells

Normalerweise sind Fremdmodelle unter dem Knoten `Objects/DeviceSet` eingefügt. Wenn ein Fremdmodell im Knoten `Objects` eingefügt war, wurde es nicht korrekt eingelesen.

Hintergrund: Die Parent-NodeId im Fremdmodell ist meistens mit „i=xxx“ angegeben. Gefunden wurde der Parent-Knoten aber nur, wenn er dem Format „ns=y;i=xxx“ entsprach. Jetzt wird nach beiden Formaten gesucht. Damit wird jetzt auch dieser Fall korrekt behandelt.

Ausserdem wird der Knoten `DeviceSet` nur dann angezeigt, wenn auch das Zusatz-Modell

<http://opcfoundation.org/UA/DI/> im Fremdmodell vorhanden ist.

1.4 RnCommTest V7.04

1.4.1 OpcUa - PubSub

1.4.1.1 Neuer Editor für PubSub-Konfigurationen

Da in verschiedenen Modulen des Tools PubSub-Konfigurationen verwaltet werden müssen (Client, Subscriber, Server), wurde ein einheitlicher Editor dafür implementiert. Das vereinfacht die Wartung/Weiterentwicklung deutlich.

Dieser Editor zeigt jetzt immer alle Einstellungen der einzelnen Konfiguration-Hierarchie-Level.

In den verschiedenen Modulen mussten deshalb diverse Funktionalitäten angepasst werden (siehe unten), welche dann aber in dieser Hilfe gesondert beschrieben sind.

1.4.2 OpcUaClient

1.4.2.1 Korrektur und Optimierung für FileTransfer

Der für dieses Tool eingesetzte Stack kann nur 1048560 Byte auf einmal lesen/schreiben. Auch ein Server kann nur eine gewisse Anzahl an Bytes auf einmal verarbeiten. Dateien werden deshalb jetzt automatisch stückweise verarbeitet, wobei die maximal unterstützte Byte-Anzahl vorher ermittelt wird (siehe OpcUaClient/Browse/[Read to local file](#)).

1.4.2.2 OpcUaClient – Attribut-Liste: Korrektur bei Wert-Anzeige als Dialog

Bei Anzeige eines Attribut-Wertes über den Kontext-Menüpunkt [Show in dialog](#) wird beim Attribut ‚Value‘ auch ein ExtensionObject korrekt aufgelöst.

1.4.2.3 MethodCall: Optimierung der Methoden-Parameter-Anzeige

Bei der Anzeige der Settings einer Methode wurden die Parameter unterteilt. So ist jetzt ersichtlich, ob es sich um einen Aufruf-Parameter oder einen Parameter zum Verhalten des Tools handelt.

1.4.2.4 MethodCall: Optimierung für abgeleitete Datentypen

Von OpcUa-Standard-Datentypen abgeleitete Datentypen für Argumente werden nun komplett unterstützt.

1.4.2.5 MethodCall: Korrektur bei der Anzeige von Ausgangs-Argumenten

Behebung eines Fehlers aus V7.03: War ein Ausgangs-Argument ein ExtensionObject, so wurde es nur als Byte-Array angezeigt, die aufgelösten Einzel-Elemente wurde nicht dargestellt.

1.4.2.6 MethodCall: Korrektur bei der Anzeige von Array-Argumenten

Bei Array-Argumenten wurde bisher in der Spalte `Description` der maximale Index angezeigt, also bei einem Array `[0..3]` wurde `xxx[3]` angezeigt. Da bei allen anderen Anzeigen (z.B. Array als Struktur-Element eines Variablen-Knotens) aber die Anzahl der Array-Elemente angezeigt wird, wird es nun hier genauso gemacht. Im obigen Fall wird also jetzt `xxx[4]` angezeigt. Damit sind alle Anzeigen einheitlich.

1.4.2.7 MethodCall: Erweiterung für Argumente mit Datentyp „Guid“

Für Argumente des Datentyps „Guid“ kann jetzt auch ein Zufallswert gesetzt werden (siehe OpcUaClient/MethodCall/InputArguments/[Set Random](#) + [Set Random All](#)).

1.4.2.8 Neuer Reiter PubSubConfigurator

In diesem Reiter können PubSub-Konfiguration eines Fremd-Geräts (z.B. B&R-Buskoppler X20BC008T) bearbeitet werden. Die Konfiguration kann sowohl von einer Datei geladen/gespeichert als auch vom Gerät hoch/runtergeladen werden. Siehe OpcUaClient/[PubSubConfigurator](#).

1.4.3 OpcUaSubscriber

1.4.3.1 OpcUaSubscriber: Anpassung an den neuen Editor

Da für die Verwaltung der Subscriber-Konfiguration der neue Editor verwendet wird, mussten folgende Anpassungen der Funktionalitäten gemacht werden:

-Die Kontextmenü-Punkte `Import Publisher configuration` und `Import Publisher configuration additionally` wurden durch `Import Publisher as Subscriber` ersetzt. Eine Konfiguration wird so immer zusätzlich importiert. Um eine einzelne Konfiguration zu importieren, kann vorher der Button `Clear` gedrückt werden.

-Der Parameter [Connection - Network Interface](#) wurde geändert. Laut der Spezifikation ist dies ein String, in dem hersteller-spezifische Angaben eingetragen werden können. Bei dieser Subscriber-Implementation kann hier optional die Ethernet-Schnittstelle angegeben werden. Um spezifikationskonform zu sein, wurde nun aus dem Auswahlfeld für die Ethernet-Schnittstelle jetzt ein String-Eingabefeld. Ist es leer (Default-Einstellung), so wird an allen Ethernet-Schnittstellen gehorcht.

Die sonstigen Funktionen bleiben kompatibel.

1.4.4 OpcUaServer

1.4.4.1 Neue Funktion: Laden von externen Modellen

Es ist jetzt möglich, externe Modelle mit dem von der Foundation spezifizierten Format `NodeSet2` zu laden (z.B. Companion Specifications oder selbst erzeugte Modelle). Die darin enthaltenen Knoten können zwar nicht wie benutzerdefinierte Knoten editiert werden, sie stehen einem verbundenen Client aber zur Verfügung. So kann bequem der Zugriff eines Clients auf Variablen oder Methoden eines Modells geprüft werden.

Siehe [Extern models](#).

Die Beispiel-Konfiguration [Example_General](#) wurde um ein selbsterzeugtes Modell erweitert.

1.4.4.2 Adressraum: Anzeige von Types-Knoten

Durch die neue Möglichkeit, externe Modelle zu importieren, ist es sinnvoll, dass auch die importierten Types angezeigt werden. So werden im Types-Ordner jetzt nicht mehr nur `EventTypes`, sondern auch `DataTypes`, `ObjectTypes`, `ReferenceTypes` und `VariableTypes` angezeigt.

1.4.4.3 Unterstützung für ComplexTypeServerFacet2017

Zum Auflösen von binär übertragenen komplexen Datentypen (also Strukturen) war bis 2017 lediglich ein Konzept namens „ComplexTypeServerFacet“ spezifiziert. Dann wurde ein zweites Konzept namens „ComplexTypeServerFacet2017“ definiert. Nähere Informationen dazu siehe [OpcUaClient/Hinweise/Auflösung von komplexen Datentypen](#).

Mit der neuen Einstellung [Complex type server facet](#) kann festgelegt werden, mit welcher Variante (oder beiden) der Server die Informationen über komplexe Datentypen veröffentlicht.

1.4.4.4 Methoden-Interpreter: Erweiterung um String-Literale mit Leerzeichen

Wird im Methoden-Code ein String als Literal, also als statischer Text angegeben, so wurden bisher Leerzeichen am Beginn und am Ende dieses Textes nicht berücksichtigt. Jetzt kann ein Text in einfache Anführungszeichen gesetzt werden, um damit auch diesen Fall abzudecken. Siehe [OpcUaServer/Server/AddressSpace/Methode: Code/Operanden](#).

Die Beispiel-Konfiguration [Example_General](#) wurde dahingehend erweitert.

1.4.4.5 Erweiterungen um Datentyp „Guid“

Es können jetzt auch Variablen-Knoten mit dem OpcUa-Datentypen „Guid“ parametrisiert werden.

Auch Methoden-Argumente dieses Typs sind jetzt erlaubt. Dazu wurden im Code-Interpreter die Operatoren `:` (Zuweisung eines anderen Arguments) und `#Random` (Zufallswert) implementiert.

Die Beispiel-Konfiguration [Example_General](#) wurde dahingehend erweitert.

1.4.4.6 File: Neuer Parameter „Access timeout“

Bei einem Knoten vom Typ `File` kann jetzt ein Timeout für das geöffnete File eingestellt werden. Erfolgt in dieser Zeit kein weiterer Zugriff seitens des Clients, wird die Datei automatisch geschlossen. Siehe [Access timeout](#).

1.4.4.7 Publisher: Anpassung an den neuen Editor

Da für die Verwaltung der Publisher-Konfiguration der neue Editor verwendet wird, mussten folgende Anpassungen der Funktionalitäten gemacht werden:

- Der Parameter `Set DataOffset automatically` wurde entfernt. Die DataOffsets werden jetzt immer neu berechnet, wenn dies nötig ist.

- Der Parameter `Set GroupVersion automatically` wurde entfernt. Die GroupVersion wird jetzt immer neu berechnet, wenn dies nötig ist.

-Der Parameter [Connection - Network Interface](#) wurde geändert. Laut der Spezifikation ist dies ein String, in dem hersteller-spezifische Angaben eingetragen werden können. Bei dieser Publisher-Implementation kann hier optional die Ethernet-Schnittstelle angegeben werden. Um spezifikationskonform zu sein, wurde nun aus dem Auswahlfeld für die Ethernet-Schnittstelle jetzt ein String-Eingabefeld. Ist es leer (Default-Einstellung), so wird an allen Ethernet-Schnittstellen gesendet. Die sonstigen Funktionen bleiben kompatibel.

1.4.4.8 Beispiel-Konfiguration um Datei erweitert

Die Beispiel-Konfiguration `Example_General` wurde um eine zweite, große pdf-Datei erweitert. Siehe [OpcUaServer/Beispiel-Konfigurationen/Example_General](#).

1.4.5 OpcUaBrMapper

1.4.5.1 Korrektur bei „Clear Settings“

Beim Zurücksetzen der Einstellungen wurde ein Parameter so gesetzt, dass beim Öffnen des Settings-Dialog eine Ausnahme auftrat. Dies ist behoben.

1.4.5.2 Optimierung für Präfixes

Es gibt die Einstellung [Ignore DV prefixes](#), welche die Verwendung von Präfixen bei den SPS-Variablen (also in der DefaultView) erlaubt. Bisher wurde dabei zum Pfad-Vergleich beim Auto-Mapping aber immer nur das letzte Element des Pfads vom Präfix befreit. Bei den beiden Properties „EngineeringUnits“ und „EURange“, welche ja Unterknoten einer Variable sind, wurde deshalb in diesem Fall kein Mapping gefunden.

Jetzt wird bei diesen beiden Properties (wenn sie in den Settings aktiviert sind, siehe [Set mappings for](#)) ebenfalls das vorletzte Pfad-Element, also der Variablenname, vom Präfix befreit. So kann auch hier automatisch ein Mapping gefunden werden.

1.5 RnCommTest V7.03

1.5.1 Allgemein

1.5.1.1 Verbesserung der Hilfe durch Links

Oft werden in Beschreibungen Verweise auf andere Kapitel angegeben. Diese wurde durch blau unterlegte Links ersetzt, welche durch einen einfachen Klick an die betreffende Stelle springen.

Beispiel: [Hinweis auf GitHub aufgenommen](#)

1.5.1.2 Hinweis auf GitHub aufgenommen

In diese Hilfe wurde unter ‚Allgemeines‘ der Hinweis auf den GitHub-Link aufgenommen, unter dem die jeweils aktuelle Version des Tools heruntergeladen werden kann.

1.5.1.3 Allgemein: Automatisches Speichern beim Beenden entfernt

Bis jetzt wurden beim Beenden des Tools automatisch verschiedene Einstellungen in verschiedenen Dateien gespeichert. Da dies auch dann geschah, wenn man zu einem Test temporär Änderungen an den Einstellungen machte und damit die vorigen Einstellungen unwiderruflich weg waren, wurde dies nun geändert. Betroffen sind:

- FrameList für Rs232, TcpClient/Master und Udp
- TextFrameList für TcpTextClient
- MbGroups für ModbusTcpMaster
- MbAdresses für ModbusTcpSlave

Achtung: Werden jetzt Änderungen an diesen Listen gemacht, müssen sie vor dem Beenden des Tools explizit gespeichert werden!

1.5.1.4 Allgemein: Rücksetzen des Layouts

Das Rücksetzen des Hauptfenster-Layouts mithilfe des Tray-Icons ist schon seit der ersten Version implementiert, war aber nicht dokumentiert. Dazu wurde ein eigener Punkt unter ‚Allgemeines‘ in diese Hilfe aufgenommen (siehe [Allgemeines/Tray-Icon](#)).

Außerdem wurde eine kleine Änderung vorgenommen: Beim Neustart des Tools wurden sowohl die X- als auch die Y-Koordinate des Hauptfensters auf 0 korrigiert, wenn sie im Minus-Bereich lagen. Dies wird jetzt nicht mehr gemacht. So startet das Tool auch dann an der letzten Position, wenn es auf einem linken Bildschirm lag.

1.5.1.5 Allgemein: Optimierung Trend

Die Trendanzeige wurde optimiert, so dass auch 64-Bit-Werte (UInt64 und Int64) korrekt angezeigt werden.

1.5.2 OpcUa

1.5.2.1 Neues Modul „OpcUaBrMapper“

Dieses neue Modul dient nicht zum Testen einer Kommunikation, sondern zum Setzen von Knoten-Referenzen für OpcUa-Fremdmodelle in einem B&R-Sps-Projekt, dem sogenannten Mapping. Es ist für Projekte ab AS4.7 einsetzbar. Siehe [OpcUaBrMapper](#).

1.5.2.2 Umstellung des OpcUa-Stacks

Es wurde von OpcUa-Stack 1.04.366.38 auf 1.04.371.50 umgestellt. Die Funktionalität des Tools sollte kompatibel zur vorigen Version sein.

1.5.3 OpcUaClient

1.5.3.1 OpcUaClient: Optimierung für falschen Attribut-Status

Beim ersten Auslesen eines Knoten wird unabhängig von der NodeClass immer die Liste aller Attribute gelesen. Ist ein Attribut aufgrund der NodeClass nicht vorhanden, sollte der Status ‚8035000=BadAttributeInvalid‘ zurückgegeben werden. Manche Server liefern aber fälschlicherweise den Status ‚80020000=BadInternalError‘. Durch die Optimierung wird in diesem Fall das Auslesen für die restlichen Attribute fortgesetzt.

Der Fehler des Servers konnte verschiedenste Auswirkungen haben, z.B. dass eine Methode nicht korrekt in die Methoden-Liste hinzugefügt werden konnte.

1.5.3.2 OpcUaClient: Unterstützung für ComplexTypeServerFacet2017

Zum Auflösen von binär übertragenen komplexen Datentypen (also Strukturen) war bis 2017 lediglich ein Konzept namens „ComplexTypeServerFacet“ spezifiziert. Dann wurde ein zweites Konzept namens „ComplexTypeServerFacet2017“ definiert. Nähere Informationen dazu siehe [Auflösung von komplexen Datentypen](#).

Mit der neuen Einstellung [Complex server facet](#) kann festgelegt werden, mit welcher Variante der Client versucht, einen komplexen Datentypen aufzulösen.

1.5.3.3 OpcUaClient: Optimierung beim Auflösen eines Datentyps

Mit dieser Optimierung wird ein seltener Fehler eines Servers umgangen (betrifft die ursprüngliche Variante „ComplexTypeServerFacet“).

Ein Server veröffentlicht die Definition von Datentypen (Strukturen oder Enumerationen) in einem sogenannten Dictionary-Datenpunkt, welcher die Definitionen als Xml enthält. Für jede enthaltene Definition hat der Dictionary-Datenpunkt normalerweise einen Unterknoten. Ein Client folgt den Verweisen vom Variablen-Datenpunkt bis zum Dictionary und kann dann mit diesen Informationen den Datentypen auflösen und so die binären Transport-Daten in den entsprechenden Datentypen oder zurück wandeln.

Leider kommt es manchmal vor, dass ein Server nicht die korrekten Unterknoten für ein Dictionary anlegt und/oder die Verweise nicht richtig gesetzt sind. Für diesen Fall durchsucht das Tool die Dictionaries auch noch textuell. Wenn nun dieser Server verschiedene Datentypen mit dem fast gleichen Namen definiert, welche sich nur durch ein Präfix unterscheiden, wurde unter Umständen das falsche Dictionary ermittelt. Da dieses aber die gewünschte Definition nicht enthält, wurde ein Fehler ausgegeben, dass der Datentyp nicht ermittelt werden konnte.

Um diesem Umstand gerecht zu werden, wurde die textuelle Suche verbessert. So kann auch in diesem Fall das korrekte Dictionary ermittelt werden.

1.5.3.4 OpcUaClient – Attribut-Liste: Wert-Anzeige als Dialog

Der Wert eines Attributs kann über den neuen Kontext-Menüpunkt ‚Show in dialog‘ bzw. Doppelklick auf das Attribut in einem eigenen Dialog angezeigt werden (siehe [Show in dialog](#)). Das ist vor allem nützlich, wenn der Attribut-Wert eine Struktur oder ein größeres Array ist.

1.5.3.5 OpcUaClient: Unterstützung des File-Transfers

Das Kontext-Menü des Browse-Baums wurde um zwei Punkte erweitert, die den File-Transfer über OpcUa unterstützen. Sie werden nur dann angeboten, wenn es sich bei dem selektierten Knoten um ein Datei-Objekt handelt.

Mit dem Befehl ‚Read to local file...‘ kann ein Datei-Inhalt vom Server gelesen und in einer lokalen Datei gespeichert werden.

Mit dem Befehl ‚Write from local file...‘ kann der Inhalt einer lokalen Datei auf den Server geschrieben werden.

Bei beiden Befehlen werden die benötigten Methoden-Aufrufen des Datei-Objekts hintereinander ausgeführt. Der Timeout für diese Aufrufe kann in den Settings gesondert eingestellt werden (siehe [File Transfer Timeout](#)).

Unterstützt werden Datei-Knoten der beiden OpcUa-Object-Typen ‚FileType‘ und ‚TemporaryFileTransferType‘.

Siehe [Read to local file](#) und [Write from local file](#).

1.5.3.6 OpcUaClient: Optimierung der Node-Browse-Dialogs

Bei der Eingabe von diversen NodeId's wird meistens ein Dialogfeld zum Browsen der Nodes angeboten. Dieser Dialog wurde optimiert.

So wird ein schon gewählter Node beim erneuten Öffnen automatisch wieder selektiert. Außerdem werden durch eine Node-Class-Filterung nur Nodes angeboten, die zur benötigten Auswahl passen.

Folgende Angaben sind betroffen:

- Wert-Anzeige: Setzen eines Datenpunkts vom Datentyp ‚NodeId‘
- DirectAccess: NodeId
- Translate: StartNodeId
- RegisterNodes: NodeId
- Subscription: Event-Filter

Siehe z.B. unter [Browse NodeId](#).

1.5.3.7 OpcUa-Client – Access.Advanced: Erweiterung für OpcUa-Datentyp

Wenn ein Struktur-Element vom Datentyp ‚Opc.Ua. ApplicationDescription‘ ist, wird es jetzt korrekt angezeigt und kann auch durch Doppelklick in einem eigenen Fenster angezeigt werden.

1.5.3.8 OpcUaClient – Subscription: Optimierung der Settings-Dialoge

Sowohl bei dem Dialog für die [Subscription Settings](#) als auch dem für die [Item Settings](#) wurden die Parameter unterteilt. So ist jetzt ersichtlich, ob es sich um einen Parameter zum Verhalten des Tools oder um einen OpcUa-Parameter handelt.

1.5.3.9 OpcUaClient – Subscription – Event-Filter: Erweiterung der Field-Auswahl

Beim Zusammenstellen des Event-Filters konnten bis jetzt nur Fields aus einer statischen Vorauswahl ausgewählt werden (z.B. ‚Severity‘). Jetzt können die Fields mit einem Browse-Dialog ausgewählt werden, welches alle aktuell am Server verfügbaren Fields anzeigt. So können nun auch Fields von selbstdefinierten Events zur Filterung verwendet werden.

Das Layout des Event-Filter-Dialogs wurde dazu geringfügig geändert. Die Filter-Einstellungen wurden in dieser Hilfe detaillierter beschrieben. Siehe [Bedingung](#).

1.5.3.10 OpcUaClient – Method Call: Korrektur für BaseDataType

Wenn ein Argument einer Methode vom Datentyp ‚BaseDataType‘ ist (also jeglichen OpcUa-Datentyp annehmen kann), so wurde bis jetzt eine Ausnahme beim Hinzufügen dieser Methode geworfen. Dies ist jetzt korrigiert.

Ist dies bei Ausgangs-Argumenten der Fall, so kann die Methode aufgerufen werden.

Ist dies bei Eingangs-Argumenten der Fall, so kann die Methode zwar aufgerufen werden, die betroffenen Argumente werden aber mit dem Wert ‚Null‘ übergeben.

Hinweis: Das Setzen eines Arguments mit Datentyp ‚BaseDataType‘ ist (noch) nicht implementiert, weil dies eine komplexe, generische Auswahl eines Datentyps erfordern würde.

1.5.3.11 OpcUaClient – Method Call: Erweiterung der Datentypen für Argumente

Einige von OpcUa-Standard-Datentypen abgeleitete Datentypen wurden für Argumente nicht unterstützt. Deshalb konnten Methoden, die Argumente von solchen Datentypen besitzen, nicht korrekt in die Methoden-Liste eingefügt werden. Dies wurde jetzt behoben. Es handelt sich dabei um Ableitungen folgender Standard-Datentypen: ByteString, DateTime, NodeId und String.

1.5.3.12 OpcUaClient – Method Call: Speichern und Laden der Methoden-Liste

Bis jetzt war es nur möglich, Argumente zu speichern und zu laden.

Jetzt ist auch ein Speichern/Laden der gesamten Methoden-Liste mitsamt den Einstellungen der Argumente (z.B. Auto Random und Random Limits) möglich. So können Test-Konfigurationen schnell wieder hergestellt werden. Siehe [Save](#).

1.5.3.13 OpcUaClient – Method Call: Ändern der Description in der Liste

Wird eine Methode hinzugefügt, so wird deren Attribut ‚Description‘ gelesen und in der zweiten Spalte der Liste angezeigt. Durch einen neuen Kontextmenü-Punkt kann diese Beschreibung geändert werden. Dies ist z.B. hilfreich, wenn eine Methode mehrmals hinzugefügt, aber mit verschiedenen Argumenten befüllt wird.

Diese benutzerdefinierte Beschreibung wird mit abgespeichert und steht beim nächsten Laden wieder zur Verfügung. Siehe [Change description](#).

1.5.3.14 OpcUaClient – Method Call: Sequentieller Aufruf der Liste

Jede Methode in der Liste kann nun durch eine Checkbox an- und abgewählt werden. Durch das Kontextmenü ist es möglich, alle angewählten Methoden durch einen Klick hintereinander ausführen zu lassen. Siehe [Call list](#).

1.5.3.15 OpcUaClient – Method Call: Setzen des FileHandles bei Datei-Methoden

Beim Aufruf der Open-Methode eines Datei-Objekts wird ein sogenanntes FileHandle als Ausgangs-Argument zurückgegeben, welches bei den folgenden Methoden (z.B. Read, Write oder Close) als Eingangs-Argument benötigt wird. Um diesen Eingangs-Wert nicht manuell setzen zu müssen, sondern automatisch aus der Open-Methode zu übernehmen, kann dies entweder manuell durch das Kontextmenü der Methoden-Liste (siehe [Set file handle for file transfer methods](#)) oder automatisiert (siehe [Set file handle for file transfer methods automatically](#)) geschehen.

1.5.3.16 OpcUaClient – Method Call: Nachträgliches Einsehen der Argumente pro Aufruf

Durch einen Doppelklick auf die Ausgabe-Zeile eines Methoden-Aufrufs im Hauptfenster können die Argumente zu diesem Aufruf nachträglich eingesehen werden. Siehe [Ausgabe im Hauptfenster](#).

1.5.3.17 OpcUaClient – OpcUaMonitor: Speichern der Filter-Einstellungen

Es werden jetzt auch die Filter-Einstellungen der Monitored- bzw. Event-Items gespeichert. So können Test-Konfigurationen schnell wieder hergestellt werden.

1.5.4 OpcUaSubscriber

1.5.4.1 OpcUaSubscriber: Import mehrerer Konfigurationen möglich

Bisher wurden beim Import bestehende Konfigurationen gelöscht. Jetzt ist es über einen neuen Kontextmenü-Punkt möglich, zu den bestehenden noch eine zusätzliche Konfiguration zu importieren. Somit können Telegramme verschiedener Publisher gleichzeitig empfangen und angezeigt werden. Siehe [Import](#).

1.5.4.2 OpcUaSubscriber: Optimierung Stop-Funktion

Wenn die Telegramme schneller als 100ms eintreffen, ist das Tool so mit der Ausgabe beschäftigt, dass es nicht mehr gleich auf Mausklicks reagiert. Der Stop-Button wurde jetzt mit einer hochpriorien Erkennung ausgestattet, so dass in jedem Fall gestoppt wird.

1.5.4.3 OpcUaSubscriber: Optimierung der Ausgabe mit neuen Optionen

Für jede als gültig erkannte NetworkMessage eines abonnierten Publishers wurde bis jetzt eine Zeile ausgegeben. Da bei einem kleinen PublisherInterval (<100ms) entsprechend viele Ausgaben gemacht werden müssen, kann zur Performance-Steigerung des Tools nun die Ausgabe über die neue Option [Network message output enabled](#) deaktiviert werden.

Außerdem gibt es jetzt die weitere neue Option [Force update rate \[ms\]](#). Ist dieser Parameter größer als 0, so wird die Wert- und Trace-Anzeige jeder ReaderGroup in diesem Intervall aktualisiert. Das ist vor allem bei einem sehr kleinen Publish-Intervall (<100ms) vorteilhaft, da das Tool dann noch anständig bedient werden kann, auch wenn es mit Meldungen überflutet wird. Der Parameter ist standardmässig auf 100ms eingestellt.

1.5.4.4 OpcUaSubscriber: Neues Dialogfeld „Statistics“

Über einen Button kann nun ein Dialogfeld eingeblendet werden, das Auskunft über die zeitlichen Intervalle der eingetroffenen Network-Messages enthält. Siehe [Statistik](#).

1.5.4.5 OpcUaSubscriber: Korrektur am Stack

Unter bestimmten Bedingungen wurden im Stack empfangene DataSetReader nicht decodiert und somit nicht an das Tool weitergeleitet. Es wurden dann keine Wert-Ausgaben gemacht.

Details: Wenn ein ReaderDataSet in einem der beiden Parameter ‚MinorVersion‘ oder ‚MajorVersion‘ auf 0 stand, wurde nicht decodiert. Dies wurde behoben.

Beim Import werden deshalb nun auch Warnungen ausgegeben, wenn einer dieser Parameter 0 ist.

1.5.4.6 OpcUaSubscriber: Korrektur/Optimierung beim Trace

Ein boolscher Wert wurde fälschlicherweise als String in den Trace eingetragen, wodurch keine klassische Bool-Linie sichtbar war, sondern eine aufsteigende Linie mit dem jeweiligen 0/1-Wert. Dies ist behoben.

Bis jetzt wurde bei jedem Erhalt eines Feld-Wertes, welches im Trace vorhanden ist, die Trendlinie um einen Wert erweitert. Dies erschwert besonders bei einem kleinen Publisher-Intervall die Analyse. Deshalb gibt es bei den Tool-Einstellungen eine neue Option ‚Add trace point only on value change‘, welche für alle Datenpunkte gilt. Ist sie angehakt, wird ein Trendpunkt nur dann eingefügt, wenn sich der Wert gegenüber dem letzten Empfang geändert hat.

1.5.5 OpcUsServer

1.5.5.1 OpcUaServer - AddressSpace: Erweiterung um Files

Im Baum können jetzt statische File-Knoten gesetzt und parametrisiert werden.

Damit kann der lesende und/oder schreibende Zugriff auf eine statische Datei gewährt werden. Siehe [File](#).

1.6 RnCommTest V7.02

1.6.1 Pvi – Update auf PviServices 4.10.1.42

Es wird jetzt die neueste Version der Bibliothek eingesetzt, bei der die korrekte Produkt-Version (V4.10.1.42) ausgelesen werden kann. Dies war bei der letzten Version (V4.9.2.9) nicht der Fall.

1.6.2 ModbusTcpMaster: Optimierung der Schreib-Ausgabe

Bei den Funktionen `WriteSingleRegister_6`, `WriteMultipleRegisters_16` und `ReadWriteMultipleRegisters_23` ist es jetzt möglich, die `InterpreteAs`-Einstellung eines `MbValue` zu ändern. So können bei der Ausgabe auch die Werte eines Schreibauftrags als andere Datentypen als `Uint16` dargestellt werden, so wie es bis jetzt nur bei den Leseaufträgen war. Mit dieser verbesserten Ausgabe ist der Wert-Vergleich zwischen Master und Slave wesentlich einfacher.

1.6.3 Zertifikats-Verwaltung: Erweiterung für Benutzer-Zertifikate

Es können jetzt auch Benutzer-Zertifikate verwaltet werden, die z.B. für OpcUa verwendet werden. Dazu gibt es den neuen Unterordner „User“, welche die Benutzer-Zertifikate enthält. In ihm können Zertifikate erstellt, ausgewählt und gelöscht werden. Je nach übergeordnetem Modul werden diese Zertifikate dann verwendet:

OpcUaClient: Ein Zertifikat kann optional für die Authentifizierung bei der Anmeldung verwendet werden.

OpcUaServer: Ein Zertifikat kann optional einem Benutzer zugewiesen werden.

Außerdem wurde der Ordner „CA“ entfernt, weil er im Tool nicht verwendet wird.

1.6.4 OpcUaClient – Server-Properties: Interner Umbau der Einstellungen

Die Verwaltung der Einstellungen wurde im Unterbau komplett überarbeitet. Damit ist die Erweiterung/Änderung von Parametern leichter und schneller umzusetzen. Dies sollte den Benutzer nicht betreffen, da die Verwaltung kompatibel zur vorigen Version sein sollte.

1.6.5 OpcUaClient – Connection: Optimierung und Erweiterung der Benutzer-Authentifizierungs-Daten-Verwaltung

Die Anzeige/Verwaltung der Benutzer-Authentifizierungs-Daten wurde überarbeitet. Statt zwei separater Auswahlfelder für Benutzername und Passwort ist jetzt eine veränderbare Liste implementiert, in der ein Benutzername samt zugehörigem Passwort gewählt werden kann.

Ausserdem ist es jetzt möglich, diese Daten nicht als Klartext, sondern versteckt als „****“ anzuzeigen, so dass kein „Schulterblick“ mehr möglich ist.

Zusätzlich kann entschieden werden, ob die Daten in der Konfigurations-Datei gespeichert werden oder ob sie nach jedem Laden neu eingegeben werden müssen.

Die zu speichernden Authentifizierungs-Daten werden ab dieser Version verschlüsselt. Geladen werden können auch alte Konfigurationsdateien, die noch die unverschlüsselten Daten enthalten.

Durch diese Massnahmen kann der Benutzer die Sicherheit der Daten gewährleisten.

1.6.6 OpcUaClient – Connection: Erweiterung um Anmeldung mit Zertifikat

Der Dialog wurde um die Möglichkeit erweitert, ein Benutzer-Zertifikat als Authentifizierung anzugeben. Das zu verwendende Zertifikat kann aus dem neuen Ordner „User“ des Tool-Stores ausgewählt werden.

1.6.7 OpcUaClient – Connection: Verwaltung der bevorzugten Sprachen

Die bevorzugten Sprachen können jetzt bequem über ein Dialogfeld verwaltet werden.

1.6.8 OpcUaClient – Connection: Auswahl des Client-Instanz-Zertifikats

Das zu verwendende Zertifikat kann jetzt bequem auch über ein Dialogfeld ausgewählt werden.

1.6.9 OpcUaClient – Connection: Benutzer-Änderung während einer Verbindung

Wird während einer Verbindung ein anderer Benutzer angemeldet, werden jetzt bessere Ausgaben gemacht.

1.6.10 OpcUaClient – MethodCall: Asynchroner Aufruf möglich

Bis jetzt wurde jeder Aufruf einer Methode synchron behandelt, d.h. das Tool war nicht bedienbar, solange der Aufruf an den Client nicht zurückgemeldet wurde, weil es auf die Rückmeldung wartete.

Jetzt ist es möglich, die Methode auch asynchron aufzurufen. Während des Aufrufs kann das Tool dann normal bedient werden. So können während eines Aufrufs einer Methode, die längere Zeit dauert, auch andere Funktionen ausgeführt werden, z.B. das Schreiben eines Knotens oder der Aufruf einer anderen Methode.

1.6.11 OpcUaClient – ServerDiag: Ausführliche Anzeige komplexer Werte

Mit einem Doppelklick auf einen Listen-Wert wird jetzt ein Dialogfeld zur Anzeige des Werts eingeblendet. Somit ist eine bequeme Analyse auch von komplexen Werten (z.B. Anmelde-History) möglich.

1.6.12 OpcUaClient – ServerDiag: Trace der Security-Werte möglich

Bis jetzt konnten nur Werte aus den Listen ‚Session‘ und ‚Subscription‘ in den Trace hinzugefügt werden. Jetzt gilt dies auch für Werte aus der Liste ‚Security‘. So kann z.B. auch die Anmelde-History im Trace verfolgt werden.

1.6.13 OpcUaClient – Monitor: Update der Session möglich

In den Eigenschaften der Verbindung des Monitors wird jetzt durch Angeben eines anderen Benutzers oder Ändern der bevorzugten Sprachen beim Drücken auf Ok der Server über die Änderung informiert. So ist ein Ummelden ohne Abbruch der Verbindung möglich.

1.6.14 OpcUaSubscriber – Diag: GroupVersion zusätzlich als Zeitstempel

Im Telegramm-Diagnose-Fenster wird jetzt eine eventuell enthaltene GroupVersion nicht nur als Zahl (entspricht den Sekunden seit 01.01.2000) angezeigt, sondern auch als Zeitstempel.

1.6.15 OpcUaServer – UserManagement: Erweiterung um Anmeldung mit Zertifikat

Jedem angelegten Benutzer kann optional auch ein Benutzer-Zertifikat aus dem neuen Ordner „User“ des Tool-Stores zugewiesen werden, mit dem er sich dann anmelden kann.

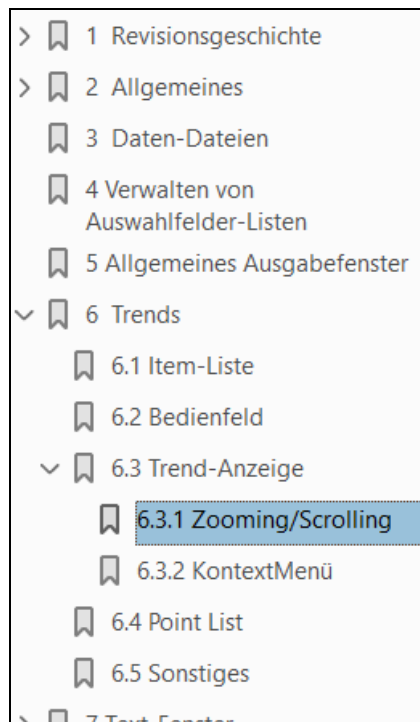
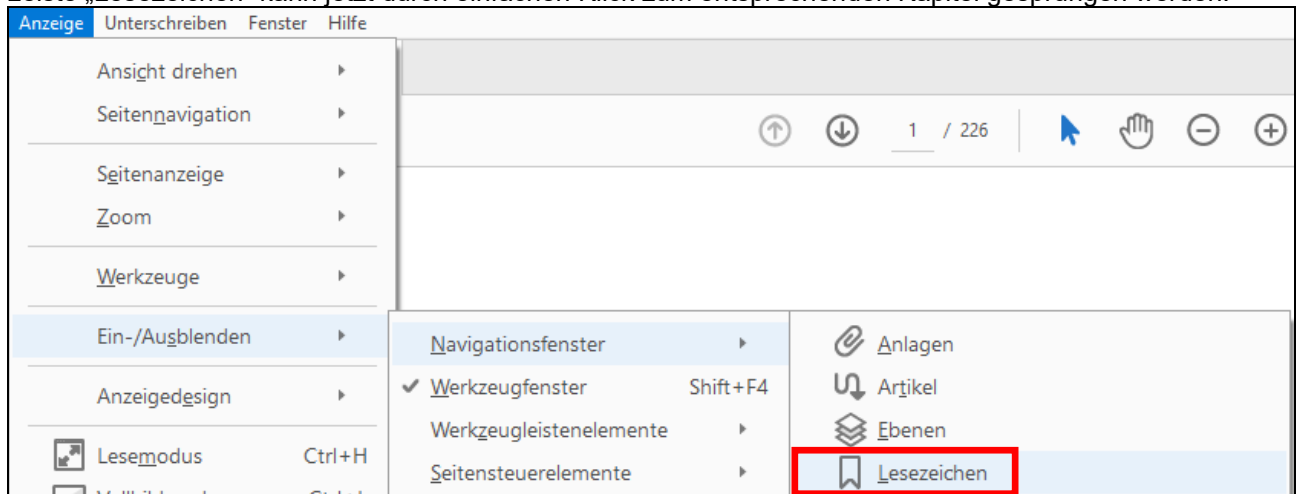
1.6.16 OpcUaServer – Ausgabefenster: Ausgaben für Clients

Im Hauptausgabefenster werden jetzt Ausgaben gemacht für Öffnen und Schließen einer Client-Verbindung sowie detailliertere Angaben zum angemeldeten Benutzer.

1.7 RnCommTest V7.01

1.7.1 Optimierung dieses Hilfe-Dokuments

Die Navigation in diesem PDF-Hilfe-Dokument wurde deutlich verbessert: Bei Anzeige der Navigations-Leiste „Lesezeichen“ kann jetzt durch einfachen Klick zum entsprechenden Kapitel gesprungen werden:



1.7.2 Zertifikate: Neue Hinweise zur Verwendung bei B&R

Der Abschnitt für die B&R-Zertifikats-Hinweise beim OpcUa-Client wurde neu unterteilt und enthält jetzt noch mehr Informationen, z.B. bezüglich CommonName und URL.

1.7.3 Umstellung des OpcUa-Stacks

Es wurde von OpcUa-Stack 1.04.365.23 auf 1.04.366.38 umgestellt. Der neue Stack enthält einige strukturelle Änderungen im Bereich PubSub. Die Funktionalität des Tools sollte dennoch kompatibel zur vorigen Version sein.

1.7.4 OpcUa-Server: Korrektur bei Zertifikats-Prüfung zur Anmeldung verschlüsselter Verbindungen

Im in der Version 7.00 eingesetzten Stack 1.04.365.23 war die Behandlung von X509-Zertifikaten geändert, speziell auch die Decodierung von Zertifikats-Extensions. Dies bedingte die Einbindung mehrerer Bibliotheken, welche in der Version 7.00 nicht eingebunden waren. Leider machte sich dies nur zur Laufzeit bemerkbar, wenn versucht wurde, eine verschlüsselte Verbindung aufzubauen. Als Folge davon wurde jede verschlüsselte Verbindung mit dem Status '0x80000000 = Bad' abgelehnt.

1.7.5 OpcUa-Client: Optimierung am Layout

Auf dem Reiter ‚OpcUaClient‘ wurde eine Layout-Optimierung vorgenommen: Der Button ‚OpcUa Monitor‘, welcher den unabhängig vom Hauptfenster arbeitenden Monitor öffnet, wurde in die Groupbox ‚Server‘ verlagert. Dies bringt einen Platzgewinn, der sich vorteilhaft auf den Rest des Reiters auswirkt.

1.7.6 OpcUaClient – Access: Neuer Reiter „RegisterNodes“

Hier können die Dienste „RegisterNodes“ und „UnregisterNodes“ zum Testen des Servers aufgerufen werden (genaue Beschreibung siehe unten).

1.7.7 OpcUa-Subscriber: Korrektur bei Datentypen ‚String‘ und ‚ByteString‘

Laut Spezifikation muss die Länge eines Telegramms einer Network-Message immer gleich sein. Aufgrund eines Fehlers im unterlagerten Stack wurde aber ein String mit der tatsächlichen Anzahl der gültigen Zeichen ausgelesen. Das führte zu falschen Werten der nachfolgenden Felder. Nun wird ein String immer mit der maximalen Anzahl der Zeichen (Parameter ‚MaxStringLength‘) ausgelesen. Dies ist jetzt auch in der Hilfe vermerkt.

1.7.8 OpcUa-Server/Publisher: Korrektur bei Datentypen ‚String‘ und ‚ByteString‘

Laut Spezifikation muss die Länge eines Telegramms einer Network-Message immer gleich sein. Aufgrund eines Fehlers im unterlagerten Stack wurde aber ein String mit der tatsächlichen Anzahl der gültigen Zeichen geschrieben. Dadurch verschob sich der Offset der nachfolgenden Felder nach vorn. Nun wird ein String immer mit der maximalen Anzahl der Zeichen (Parameter ‚MaxStringLength‘) geschrieben. Dies ist jetzt auch in der Hilfe vermerkt.

1.7.9 OpcUa-Server/Publisher: Korrektur bei Datentyp ‚String‘

Bei einem Feld vom Typ ‚String‘ konnte der Parameter ‚MaxStringLength‘ nicht verändert werden, was aber unbedingt notwendig ist, da der Wert nicht automatisch ermittelt werden kann. Siehe dazu auch den überarbeiteten Abschnitt zu diesem Parameter.

1.7.10 OpcUa-Server/Publisher: Korrektur bei Datentyp ‚DateTime‘

Bei einem Feld vom Typ ‚DateTime‘ wurde beim Auswählen der SourceNode eine Fehlermeldung ausgegeben, dass nur numerische und String-Variablen erlaubt sind. Jetzt werden auch DateTime-Werte erlaubt.

1.7.11 OpcUa-Subscriber: Erweiterung bei Diagnose

Zur besseren Analyse des Telegramms werden jetzt der Startoffset, der Endoffset und die Anzahl der markierten Bytes angezeigt.

Außerdem wird jetzt eine Warnung ausgegeben, wenn zu wenig Bytes für alle Felder vorhanden sind.

1.7.12 ModbusTcpSlave: Optimierung der Ausgabe

Wenn ein Holding-Register, das ja aus 16 Bit besteht, als UInt8 interpretiert wird, wurde bis jetzt bei der Ausgabe der angefragten Adressen immer nur das erste Byte ausgegeben. Jetzt werden beide Bytes ausgegeben.

Bei Interpretierung als Int8 wurden bis jetzt aufgrund eines Fehlers keine Werte ausgegeben. Auch hier werden jetzt beide Bytes ausgegeben.

1.7.13 ModbusTcpSlave: Korrektur beim Auswerten von Telegrammen

Es wurde ein Fehler korrigiert, der nur unter bestimmten, seltenen Fällen auftritt.

Wurden mehrere Anfragen in einem Telegramm empfangen, so wurde nur die erste darin enthaltene Anfrage beantwortet. Da manche Master, besonders bei kurzen Intervallen, es so machen, wurde der Slave daraufhin korrigiert. Jetzt werden alle Anfragen in einem Telegramm erkannt und beantwortet.

Zu diesem Thema wurde auch der neue Abschnitt ‚Hinweise zu Ethernet‘ aufgenommen.

1.8 RnCommTest V7.00

1.8.1 Ethernet: Erweiterung beim Auslesen der Adapter

Es wird jetzt zusätzlich der Name der Schnittstelle ausgegeben (wie in Netzwerkverbindungen unter Windows angezeigt). Dies erleichtert das Erkennen erheblich.

1.8.2 Umstellung des OpcUa-Stacks

Es wurde von OpcUa-Stack 1.04.363.107 auf 1.04.365.23 umgestellt.

1.8.3 Neues Modul OpcUa-Subscriber

Es wurde ein OpcUa-Subscriber implementiert, der frei parametrierbar ist und mit dem PubSub-Meldungen empfangen werden können. Es können auch Publisher-Konfigurationen von anderen Teilnehmern (z.B. Automation Studio) importiert werden.

Besonders hilfreich ist die umfangreiche Diagnose-Funktion, mit der herausgefunden werden kann, warum eine empfangene Meldung nicht korrekt interpretiert wird und welche Parameter wie geändert werden müssen.

1.8.4 OpcUa-Server: Neuer Reiter 'Publisher'

Auf diesem Reiter können Knoten-Werte aus dem angelegten Adressraum über einen OpcUa-Publisher durch PubSub-Meldungen versendet werden. Publisher-Konfigurationen können ex- und importiert werden. Dadurch ist ein Austausch mit anderen Teilnehmern (z.B. RnCommTest-Subscriber oder Automation Studio) möglich.

1.8.5 OpcUa-Server: Neue Funktionalität 'ReverseConnect'

Der Server kann jetzt ReverseConnect-Einladungen an einen Client schicken, damit sich dieser dann automatisch auf den Server verbinden kann.

1.8.6 OpcUa-Client: Neue Funktionalität 'ReverseConnect'

Der Client kann jetzt auf ReverseConnect-Einladungen eines Servers horchen und sich dann automatisch verbinden.

1.8.7 OpcUa-Client: Korrektur bei Ermittlung von Struktur-Definitionen

Wenn es zwei gleichnamige Datentypen mit verschiedenem Gültigkeitsbereich seitens des Servers gab, wurde bis jetzt der Datentyp nicht eindeutig aufgelöst oder nicht gefunden, weil den Referenzen nicht bis zum Ende gefolgt wurde. Folge davon waren falsche Darstellung bis hin zu einer Exception.

Jetzt wird durch Nachverfolgung bis zur 'HasDescription'-Referenz der eindeutige Name ermittelt und so auch der richtige Datentyp aufgelöst.

Aufgefallen ist dies bei AS4.9, denn hier werden neuerdings zur Unterscheidung von lokal definierten Datentypen die Tasknamen bei der Vergabe der Datentyp-Namen mitverwendet (AS-Revisionsgeschichte ID#400295663).

1.8.8 OpcUa-Client: Optimierung beim Auslesen von Attributen

Jeder Knoten einer bestimmten NodeClass besitzt eine spezifizierte Liste an zugehörigen Attributen (z.B. hat ein 'Variable'-Knoten das Attribut 'Value', ein 'Object'-Knoten hingegen nicht).

Wenn ein Server eine falsche Attribut-Liste zu einem Knoten einer bestimmten NodeClass anbietet, wurde bis jetzt jedes Attribut (auch die gültigen) als nicht gültig angezeigt. Natürlich ist dies ursächlich ein Fehler des Servers, trotzdem wird es jetzt am Client abgefangen und nur die zugehörigen und damit gültigen Attribute angezeigt.

1.8.9 OpcUa-Client: Korrektur bei zurückgewiesenen Zertifikaten

Wurde ein Server-Zertifikat zurückgewiesen (automatisch oder manuell), so wurde dieses fälschlicherweise im Ordner '.../OpcUa/Server/Certificates/Rejected/certs/' abgelegt. Jetzt wird es korrekt im Ordner '.../OpcUa/Client/Certificates/Rejected/certs/' abgelegt.

1.8.10 Pvi – Korrektur bzw. Erweiterung beim Service-Browsing

Seit V5.04 funktionierte das Browsen der Zielsysteme im Netzwerk über SNMP nicht mehr. Grund hierfür war ein Bug in der Bibliothek „PviServices“ in der Version V4.9.2.9. Jetzt wird die neuere Version V4.9.3.20 verwendet, in der die Browse-Funktion wieder korrekt arbeitet (unabhängig von der installierten Pvi-Version).

Während des Browsens wird jetzt auch ein Laufbalken angezeigt.

Wird auf eine Zeile in der Liste der gebrowsten Cpu's doppel-geklickt, so öffnet sich nun ein Dialogfeld, das alle beim Browsen empfangenen Informationen anzeigt.

Außerdem wurde der für das Cpu-Browsing wichtige Hinweis auf die SPS-Einstellung „SNMP aktivieren“ in diese Hilfe aufgenommen.

1.8.11 Pvi – Veränderbare Item-Größe

Der Bereich zum Anzeigen des angewählten Items kann jetzt in der Größe variiert werden. So sind Listen sehr viel bequemer einsehbar.

1.8.12 Pvi – Neue Einstellung ‚Log ValueChange events‘

Durch diese neue Checkbox wird im Ausgabe-Fenster bei Empfang der Änderungs-Nachricht einer Variablen eine Zeile eingetragen.

Zusätzlich werden Fehler beim Variablen-Handling immer ausgegeben.

1.8.13 Pvi – Neue Angaben beim Object ‚Service‘

Wird im Browse-Baum das Object ‚Service‘ angeklickt, so werden jetzt zusätzlich einige Versions-Informationen ausgelesen und angezeigt.

1.8.14 Pvi – Mehr Cpu-Informationen

Der Baum mit Informationen, die von der Cpu gelesen werden können, wurde erheblich erweitert. Dazu gibt es jetzt den neuen Knoten ‚CpuInfos‘ mit etlichen Unterknoten.

1.8.15 Pvi – Optimierung bei Logbuch

Das Auslesen und die Anzeige wurden überarbeitet.

So wird jetzt versucht, die Beschreibungen des neuen Formats (EventId) zu ermitteln.

Außerdem gibt es eine Textbox, aus der Daten des aktuell selektierten Eintrags bequem in die Zwischenablage kopiert werden können.

Die Optimierungen betreffen auch das Fenster der Detailansicht.

1.8.16 Pvi – Neue Angaben beim Object ‚Hardware‘

Wird im Browse-Baum das Object ‚Hardware‘ angeklickt, so kann jetzt über einen Button die Xml-Ausgabe angezeigt werden. Sie enthält die über die Hardware-Module hochgelesenen Informationen als Klartext.

1.8.17 Pvi – Korrektur bei der Trendanzeige

Wurde ein IoDatapoint angeklickt, der sich nicht ändert, wurde kein Anfangs-Punkt im Trend eingefügt.

Dann war zwar eine Kurve vorhanden, aber es wurde kein Punkt dargestellt.

Jetzt wird beim Anklicken automatisch der aktuelle Wert in den Trend hinzugefügt, so dass auch bei Nicht-Änderung wenigstens ein Punkt sichtbar ist.

1.9 RnCommTest V6.01

1.9.1 Trend: Korrektur

Wenn mehrere Kurven angezeigt wurden, aber eine davon keine Punkte enthielt, dann wurde die X-Skalierung falsch berechnet. Als Folge davon wurden alle Punkte der restlichen Kurven am linken Rand dargestellt. Dies ist jetzt behoben.

1.9.2 Udp: Erweiterungen

Für den eigenen Socket und den der Gegenstelle können jetzt verschiedene Ports eingestellt werden. Die Advanced-Einstellung ‚DontRoute‘ wurde eliminiert, weil sie nur bei Tcp-Verbindungen ausgewertet wird. Dafür gibt es die neue Advanced-Einstellung ‚NoDelay‘, welche auch bei Udp-Verbindungen gültig ist.

Desweiteren wurde die Möglichkeit geschaffen, Telegramme an bis zu 4 Multicast-Adressen mitzuhören und auszugeben.

1.9.3 Umstellung des OpcUa-Stacks

Es wurde von OpcUa-Stack V1.04.359.31 auf 1.04.363.107 umgestellt.

1.9.4 OpcUa – Zertifikats-Erstellung: Korrektur + neue Angaben

Beim Erstellen eines neuen Zertifikats können jetzt das Start-Datum der Lebensdauer sowie der Hostname bzw. die IP-Adresse angegeben werden.

Außerdem wurde die Einstellung ‚sha256RSA‘ bei ‚SignatureAlgorithm‘ (sha1RSA, sha256RSA) nicht beachtet, sondern immer mit ‚sha1RSA‘ erstellt. Dies ist jetzt behoben. Zudem wurde die neue Einstellung ‚sha512RSA‘ hinzugefügt und die Schlüssellänge standardmässig statt von 1024 auf 2048 Bits gestellt. Bei der Anzeige des Zertifikats-Inhalts wird jetzt auch die Schlüssellänge angegeben.

Da das bis jetzt für Client und Server verwendete Tool-Zertifikat (Thumbprint = ‚EFE0E626B015436977BAE6EBDD22F2F2130AF4DB‘) bereits abgelaufen war, wurde für dieses Release ein neues erstellt (Thumbprint = ‚AAAF0A8F345BEB15A0CFA42E4B5BE8871B81456C‘).

Es hat folgende Merkmale:

- Gültigkeit von 01.01.2020 bis 01.01.2120, also noch 100 Jahre
- SignatureAlgorithm ‚sha256RSA‘
- Schlüssellänge 2048

Die letzten beiden Einstellungen sind unbedingt nötig bei Verwendung von vertrauenswürdigen Listen mit Policy ‚Basic256Sha256‘.

Achtung: Wenn an zu testenden Gegenstellen vertrauenswürdige Listen verwendet werden, müssen diese möglicherweise an das neue Zertifikat angepasst werden!

Außerdem wurde das neue Kapitel „OpcUaClient/Certificates/Hinweise zur Verwendung von Zertifikaten in der vertrauenswürdigen Liste bei B&R“ eingefügt, welches erklärt, bei welcher Security-Policy welche Zertifikats-Informationen zur Gültigkeit des Zertifikats beitragen.

1.9.5 OpcUa – Zertifikats-Verwaltung: Neue Funktion ‚Show certificate from hex string‘

Der neue Kontext-Menüpunkt ‚Show certificate from hex string‘ erlaubt die Eingabe eines Hex-Strings und wandelt diesen in ein Zertifikat, dass dann angezeigt wird. Dies ist vor allem dann nützlich, wenn Telegramme mit Netzwerk-Diagnose-Tools wie z.B. Wireshark aufgezeichnet werden, die die darin als Byte-Array enthaltenen Zertifikate als Hex-String darstellen.

1.9.6 OpcUa-Client: Optimierung bei Knoten-Browse-Dialog

Jedesmal, wenn der Browse-Dialog geöffnet wurde, war der Baum zugeklappt. Jetzt ist der zuletzt ausgewählte Knoten selektiert.

1.9.7 OpcUaClient – Server-Properties: Einstellung ‚Replace discovered hostname with IP‘

Diese Einstellung gibt es, weil laut OpcUa-Spezifikation ein Server beim Verbindungsaufbau standardmässig nicht die IP-Adresse im Endpoint zurück liefert, sondern den Hostnamen (auch wenn die Anfrage über eine IP-Adresse erfolgte). Wenn dieser Hostname nicht aufgelöst werden kann (z.B. weil kein DNS-Server verfügbar ist), kann keine Verbindung aufgebaut werden. Durch diesen Haken wird der gelieferte Hostname mit der IP-Adresse des gesuchten Servers ersetzt und damit dieses Problem umgangen.

Der in der Version 6.00 verwendete OpcUa-Stack V1.04.359.31 ersetzte diesen Hostnamen automatisch. Somit hatte diese Einstellung keinen Einfluss mehr.

Der OpcUa-Stack wurde jetzt so abgeändert, dass diese Einstellung wieder so wie vorher funktioniert.

1.9.8 OpcUaClient – Server-Properties: Geänderte Einstellung ‚Browse sort mode‘

In der letzten Version wurde der neue Parameter ‚Browse sort mode‘ eingeführt, mit der die Sortierung der Knoten im Browse-Baum verändert werden konnte.

Die Einstellung ‚MethodsFirst‘ wurde in ‚MethodsVariablesObjects‘ umbenannt und berücksichtigt jetzt auch andere Knoten-Arten. Sortierung:

- Methoden
- Variablen
- Objekte und Typen

Die Einstellung ‚Alphabetical‘ funktioniert wie bisher.

1.9.9 OpcUaClient – Server-Properties: Neue Einstellung ‚Clear subscription lists on connect‘

Die Ausgabe-Liste(n) auf dem Reiter ‚Subscription‘ wurden bis jetzt nicht automatisch gelöscht, damit sie auch nach Verbindungsabbruch oder einem manuellen Disconnect/Connect noch vorhanden sind und somit noch analysiert werden können. Sie mussten bei Bedarf jedesmal per Kontextmenü gelöscht werden.

Ist diese neue Einstellung aktiviert, so werden bei jedem manuellen Connect die Ausgabe-Listen gelöscht, damit diese nicht zu unübersichtlich werden.

1.9.10 OpcUaClient – Server-Properties: Neue Einstellungen für EventItems

Beim Abonnieren eines EventItems werden standardmässig alle Felder aller Events abonniert (aber nur diejenigen angezeigt, die nicht ‚Null‘ sind).

Manche Server zeigen allerdings ein seltsames Verhalten, wenn auch die Felder der Gruppe ‚AuditSecurityEventType‘ (i=2058) abonniert werden (z.B. wurde die Verbindung beendet und nicht mehr zugelassen). Um diesem Fehlverhalten einiger Server auszuweichen, wurden die Felder dieser Gruppe nicht automatisch hinzugefügt.

Beim Untersuchen mancher Fälle sind aber genau diese Felder interessant. Um diese Felder dann nicht manuell hinzufügen zu müssen, dennoch dem obigen Umstand Rechnung zu tragen, wurden neue Einstellungen implementiert. Mit diesen ist es möglich, das automatische Hinzufügen für bestimmte Unter-Event-Typen zu aktivieren. Durch einen Doppelklick auf den Eintrag ‚EventItem Settings‘ öffnet sich ein Dialogfeld, in dem die Unter-Event-Typen von ‚AuditSecurityEventType‘ einzeln aktiviert werden können:

- AuditCertificateEventType fields
- AuditChannelEventType fields
- AuditSessionEventType fields

Standardmässig sind alle drei nicht angehakt, so dass das Verhalten kompatibel zu vorigen Versionen ist.

1.9.11 OpcUaClient – Server-Properties: Neue Einstellung ‚Do not disconnect when tool is closed‘

Normalerweise wird beim Beenden des Tools auch eine eventuell bestehende OpcUa-Client-Verbindung korrekterweise abgemeldet (Disconnect).

Mit dieser Einstellung kann dieses Verhalten geändert werden. Damit lassen sich auch Fälle reproduzieren, bei denen sich ein Client nicht korrekt abmeldet (wie z.B. bei Spannungsverlust eines SPS-Clients). Für den Normalfall sollte diese Einstellung auf False belassen werden.

1.9.12 OpcUaClient – Server-Properties: Neue Einstellung ‚Advanced – Trim slash at end of endpoint URL‘

Eine EndpointUrl kann einen abschliessenden Slash „/“ haben oder nicht. Normalerweise sollte dies keinen Einfluss auf das Verhalten eines Servers haben. Um dies testen zu können, kann nun die Verwendung des Slashes gesteuert werden.

Für den Normalfall kann diese Einstellung auf False belassen werden.

1.9.13 OpcUa-Client – Connection: Optimierung der Fehlerausgabe bei Verbindungsaufbau

Tritt beim Verbindungsaufbau ein Fehler auf, so werden nun bessere Ausgaben gemacht, z.B. wird der OpcUa-Statuscode mit ausgegeben. Dies vereinfacht die Analyse der Ursache.

1.9.14 OpcUaClient – Connection: Optimierung beim Aufbauen einer Verbindung

Beim Aufbauen einer Verbindung werden zuerst die unterstützten Endpoints vom Server abgefragt und anschliessend mit dem gewünschten Endpoint des Clients verglichen. Der passende Endpoint wird dann zur Verbindung verwendet. Bis jetzt wurden dabei nur die Security-Einstellungen (Policy und Mode) verglichen.

Laut OpcUa-Spezifikation liefert ein Server hier standardmäßig nicht die IP-Adresse in den Endpoints zurück, sondern den Hostnamen (auch wenn die Anfrage über eine IP-Adresse erfolgte, siehe Punkt Einstellung ‚Replace discovered hostname with IP‘). Wenn dieser Hostname nicht aufgelöst werden kann (z.B. weil kein DNS-Server verfügbar ist), kann keine Verbindung aufgebaut werden. Wenn ein Client nun diesen Hostname nicht selbstständig durch die IP-Adresse ersetzen kann (was bei Clients einiger Hersteller durchaus vorkommt), kommt keine Verbindung zustande.

Um dieses Manko mancher Clients zu umgehen, ist der B&R-Server ab AS4.8 toleranter gestaltet und sendet einen Endpoint unter gewissen Umständen zweimal (siehe-B&R Hilfe): mit Hostnamen und mit IP-Adresse.

Deshalb wird nun im RnCommTest auch die EndpointUrl zum Vergleich herangezogen. Ist ein Endpoint vorhanden, der die gleiche EndpointUrl hat (also z.B. mit IP-Adresse), wird dieser bevorzugt. Wird hier keiner gefunden, wird der Endpoint verwendet, der zumindest die gewünschten Security-Einstellungen unterstützt (wie vorher).

Somit wird der am besten passende Endpoint (mit Hostname oder IP-Adresse) verwendet.

1.9.15 OpcUaClient – Server-Properties: Korrektur bei Änderung des Benutzers

Bei einer bestehenden Verbindung sollte der Benutzer ohne Disconnect geändert werden können. Leider funktionierte dies nicht mehr. Die Änderung des Benutzers wurde nicht übernommen. Das ist jetzt korrigiert.

Auch das Verwalten der Liste (Reihenfolge ändern usw.) wurde korrigiert.

1.9.16 OpcUa-Client – Access.Advanced: Anzeige von Byte() als Zertifikat

Da ein Zertifikat bei OpcUa als Byte-Array übertragen wird, kann jetzt ein Byte-Array durch den Kontext-Menüpunkt ‚Show as certificate‘ auch als Zertifikat interpretiert und angezeigt werden.

Damit das Kontext-Menü nicht zu groß und unübersichtlich wird, wurden zudem wurden alle ‚Show as xxx‘-Menüpunkte im Punkt ‚Show as‘ zusammengefasst.

1.9.17 OpcUa-Client – Access.Advanced: Korrektur bei ExtensionObjects

Bei einem Wert, der als Byte-Array in einem ExtensionObject (also z.B. eine Struktur) übertragen wird, wird als erster Eintrag das übertragene Byte-Array mit dessen Länge angezeigt. Diese Anzeige wurde in manchen Fällen nicht aktualisiert, wenn sich die Länge geändert hat. Dies ist nun korrigiert.

1.9.18 OpcUa-Client – Access.Advanced: Optimierung bei Enumerationen

Wenn eine Enumerationsvariable, die ja eigentlich ein Int32 ist, einen Wert enthielt, der nicht durch die Enumeration abgedeckt war, so wurde bisher immer der erste gültige Wert der Enumeration angezeigt. Die Anzeige spiegelte also nicht den korrekten Wert wider.

Jetzt wird der tatsächliche Wert als Int32 angezeigt und kann dadurch auch richtig gesetzt werden.

1.9.19 OpcUa-Client – Access.Advanced: Anzeige von String-Collections

Wenn ein Struktur-Element vom Datentyp ‚Opc.Ua.StringCollection‘ ist, wird es jetzt korrekt angezeigt und kann auch durch Doppelklick in einem eigenen Fenster angezeigt werden.

1.9.20 OpcUaClient - Subscription: Korrektur bei der Trendanzeige

Wurde eine Variable abonniert, die sich nicht ändert, wurde kein Anfangs-Punkt im Trend eingefügt. Dann war zwar eine Kurve vorhanden, aber es wurde kein Punkt dargestellt.

Jetzt wird beim Abonnieren automatisch der aktuelle Wert gelesen und in den Trend hinzugefügt, so dass auch bei Nicht-Änderung wenigstens ein Punkt sichtbar ist.

1.9.21 OpcUaClient - Subscription: Implementierung für Alarms & Conditions

Events, die vom Typ ‚ConditionType‘ oder darunter sind (also Konditionen und Alarme), werden jetzt zusätzlich besonders behandelt. Sie werden in speziellen Listen angezeigt und die dafür notwendigen Methoden können sehr bequem aufgerufen werden.

1.9.22 OpcUaClient - Subscription: Auflösung von Event-Feldern mit komplexem Datentyp

Wenn ein Event-Feld den Datentyp ‚BaseDataType‘ hat, kann es alle OpcUa-Datentypen aufnehmen (also z.B. einen Double, einen String oder eine benutzerdefinierte Struktur, siehe Feld ‚Context‘ des Events ‚ProgressEventType‘).

Mit einem Doppelklick auf die Ausgabe eines empfangenen Events wird die Feld-Liste ausgegeben. Mit einem Doppelklick auf ein Feld wird das Feld dann ausführlich angezeigt. Hier werden jetzt auch komplexe Datentypen aufgelöst.

1.9.23 OpcUaClient - Subscription: Erweiterung beim automatischen Abonnieren von Event-Feldern

Bis jetzt wurden nur Felder abonniert, die ein direkter Unterknoten eines Event-Typs sind.

Ein Event-Typ kann aber auch ein Element der NodeClass ‚Object‘ haben, das wiederum Felder enthält. So können bestimmte Felder gruppiert werden. Auch diese Felder werden jetzt automatisch abonniert.

Für Alarm&Conditions wurde eine Erweiterung notwendig: Es wird jetzt automatisch auch der Wert ‚ConditionId‘ abonniert. Es handelt sich dabei um eine Information, die nicht als klassisches Feld aufliegt, aber trotzdem abonniert werden kann (siehe ‚Hinweise zu Alarm&Condition-Events‘).

Beim Hinzufügen oder Entfernen von Items in der Subscription wurden zusätzlich Optimierungen in der Performance gemacht. So wird ein Item nicht mehr erst angelegt und dann die Parameter geändert, sondern das Item wird gleich mit den richtigen Parametern angelegt. So werden der Client, das Netzwerk und auch der Server weniger belastet.

1.9.24 OpcUaClient - MethodCall: Unterstützung weiterer OpcUa-Datentypen

Argumente der OpcUa-Datentypen Counter, IntegerId, VersionTime und BitFieldMaskDataType werden jetzt korrekt unterstützt.

1.9.25 OpcUaClient – Historical Access: Neue Fehlerausgabe

Wenn ein Node in den Historical Access hinzugefügt wird, der Benutzer aber keine Zugriffs-Rechte hat, wird jetzt eine Fehlermeldung ausgegeben.

1.9.26 OpcUaClient – ServerDiag: Korrektur bei Stop

Wurde eine Diagnose gestoppt, wurden die Einträge in der Liste ‚Security‘ nicht gelöscht. Nach einem erneuten Start wurden dann eventuell noch alte Werte angezeigt. Deshalb wird diese Liste jetzt bei Stop gelöscht.

1.9.27 OpcUaClient - Monitor: Optimierung bei Event-Items

Diese Änderung betrifft nur den OpcUa-Monitor, der ja unabhängig vom Hauptfenster betrieben werden kann und bei dem die Item-Liste der Subscription gespeichert und geladen werden kann.

Bei einem Event-Item werden jetzt automatisch alle Felder abonniert (wie es auch beim Hauptfenster ist).

1.9.28 OpcUaServer – Settings: Neuer Parameter ‚Correct client signature algorithm‘

In bestimmten Situationen kann dadurch ein Fehler mancher Clients bei Verwendung der Security-Policy ‚Basic256Rsa256‘, ‚Aes256_Sha256_RsaOaep‘ oder ‚Aes256_Sha256_RsaPss‘ umgangen werden. Genaueres dazu ist in der Beschreibung des neuen Parameters zu lesen.

1.9.29 OpcUaServer – Diag: Ausgabe des Client-Zertifikats

In der Ausgabe-Liste der selektierten Session wird jetzt auch der Thumbprint des Client-Zertifikats angezeigt. Mit Doppelklick darauf wird das komplette Zertifikat angezeigt.

1.10 RnCommTest V6.00

1.10.1 Umstellung der Entwicklungsumgebung MS Visual Studio

Es wurde von Visual Studio 2015 auf 2019 umgestellt.

1.10.2 Umstellung des Dot.Net-Frameworks

Es wurde von Dot.Net-Framework 4.5.1 auf 4.7.2 umgestellt.

1.10.3 Layout-Anpassungen für Windows10

Seltsamerweise funktioniert die automatische Skalierung unter Windows10 anders als in Windows7. Deshalb wurden einige Elemente in Größe und Position überarbeitet, um auch in Windows10 angenehm dargestellt zu werden.

1.10.4 Umstellung des OpcUa-Stacks


Es wurde von OpcUa-Stack V1.03.342 auf V1.04.359.31 umgestellt. Dadurch konnten neue Funktionalitäten implementiert werden.

1.10.5 OpcUaClient/Server: Unterstützung neuer Security-Policies

Es sind folgende Policies hinzugekommen:

- Basci256Sha256
- Aes128_Sha256_RsaOaep
- Aes256_Sha256_RsaPss

1.10.6 OpcUaClient – Server-Properties: Neue Einstellung ‚Evaluate sub nodes on browsing‘

Bis jetzt wurden beim Browsen eines Knotens immer die Unterknoten ermittelt, um das Aufklapp-Zeichen  nur bei Knoten anzuzeigen, die auch tatsächlich Unterknoten haben.

Jetzt gibt es dazu den neuen Parameter ‚Evaluate sub nodes on browsing‘. Wenn aktiviert (=Standardeinstellung), verhält es sich wie bisher. Wenn nicht aktiviert, wird das Aufklapp-Zeichen bei jedem Knoten angezeigt, auch wenn er keine Unterknoten enthält. Wird ein solcher Knoten aufgeklappt, verschwindet das Aufklapp-Zeichen. Bei Knoten mit vielen Unterknoten ist das Browsen dann viel performanter.

1.10.7 OpcUaClient – Server-Properties: Neue Einstellung ‚Browse sort mode‘

Bis jetzt wurden bei Sortierung von Knoten im Browse-Baum die Methoden immer am Anfang einsortiert. Jetzt gibt es den neuen Parameter ‚Browse sort mode‘. Die Einstellung ‚MethodsFirst‘ verhält sich wie bisher, die Einstellung ‚Alphabetical‘ sortiert auch die Methoden alphabetisch ein.

1.10.8 OpcUaClient: Optimierung bei Auflösung von Datentypen

Wenn mehrere Modelle in einem Server verwendet werden, kann es sein, daß zwei unterschiedliche Datentypen den gleichen Namen haben. Sie werden dann nur über die NodeId unterschieden.

Dieser Fall wird jetzt auch unterstützt, indem auch die NodeId des Datentyps (wenn vorhanden) zur Ermittlung der Datentyp-Beschreibung verwendet wird.

1.10.9 OpcUaClient: Überarbeitung des Wert-Steuerlements

Das Steuerelement zum Anzeigen/Editieren eines Wertes (z.B. bei Access.Advanced oder Subscription) wurde überarbeitet. Durch diese Massnahme ist eine zukünftige Implementierung neuer OpcUa-Datentypen einfacher. Außerdem ist der Aufbau der Anzeige in manchen Fällen optimiert worden, so dass dies schneller geht.

Zusätzlich werden jetzt bei Struktur-Arrays die einzelnen Elemente der einzelnen Strukturen aufgelöst.

Es werden jetzt eine ganze Reihe von zusätzlichen OpcUa-Datentypen unterstützt.

Außerdem wurden die Kombinationen aller implementierten Datentypen für Einzel-Wert, Arrays (ein- und mehrdimensional), Strukturen sowie Struktur-Arrays (ein- und mehrdimensional) getestet und etwaige Fehler eliminiert.

Auch das Einfügen oder Löschen von Elementen bei dynamischen Arrays wurde erweitert und ausgiebig getestet.

Zudem wird bei einem NodeId-Wert auch der zum NamespaceIndex gehörende NamespaceUri ausgegraut angezeigt.

1.10.10 OpcUaClient - Subscription: Anzeigen des Wertes in eigenem Dialog

Durch den Kontextmenü-Punkt ‚Show value in dialog‘ kann ein Wert jetzt auch in einem eigenen Fenster angezeigt werden. Die ist z.B. bei sehr langen Strings nützlich.

1.10.11 OpcUaClient – MethodCall: Korrektur bei dynamischen Array-Argument

Wenn ein Eingangs-Argument ein dynamisches Array ist, wurde nur ein skalar Element angelegt. Somit konnte die Anzahl der Elemente auch nicht verändert werden. Der Aufruf endete dann mit dem Status 0x80A1000 = „BadInvalidArgument“.

Jetzt wird das Argument tatsächlich als Array mit einem Element erzeugt.

1.10.12 OpcUaClient – MethodCall: Korrektur bei String-Array-Argument

Wenn ein Eingangs-Argument ein String-Array ist, wurde nur ein skalar Element angezeigt. Jetzt wird das Argument tatsächlich als Array angezeigt.

1.10.13 OpcUaClient – Certificates: Längerer Gültigkeitsbereich möglich

Beim Erstellen eines neuen Zertifikats konnte bis jetzt nur ein Gültigkeitszeitraum von max. 120 Monaten (=10 Jahre) eingegeben werden. Jetzt wurde das Limit auf 1200 Monate (=100 Jahre) gesetzt.

1.11 RnCommTest V5.04

1.11.1 Layout-Anpassungen für Windows10

Seltsamerweise funktioniert die automatische Skalierung unter Windows10 anders als in Windows7. Deshalb wurden einige Fenster in Größe und Layout überarbeitet, um auch in Windows10 angenehm dargestellt zu werden.

1.11.2 Neuer Dialog zum Anzeigen/Verändern von Grafiken (Image)

Zum Anzeigen und Verändern von Grafiken (Images), die als Byte-Array gespeichert sind, wurde ein eigener Dialog implementiert.

1.11.3 Ethernet

Das Modul wurde überarbeitet. Aus Platzgründen wurde das Modul in mehrere Tabs aufgeteilt.

Es werden jetzt alle Ethernet-Adapter (auch jene ohne gültige IP-Adresse) ausgelesen und auch mehr Details angezeigt. So wird das Erkennen eines Adapters stark erleichtert.

Das Pingen kann jetzt auch zyklisch ausgeführt werden. Zusätzlich kann auch die Anzahl der Daten-Bytes angegeben werden. Die Ausgabe erfolgt in einem eigenen Ausgabe-Fenster, optional zusätzlich im Haupt-Ausgabe-Fenster.

Außerdem wurde der neue Reiter „TrafficGenerator“ implementiert. Mit diesem kann künstlicher UDP-Datenverkehr erzeugt werden.

1.11.4 ModBusTcp Slave – Erweiterte Ausgabe der Daten

Im Fenster ‚Data‘ werden die Telegramme und die damit adressierten Register ausgegeben. Der Register-Wert wird als Hex- und Dezimal-Zahl ausgegeben. Nun wird zusätzlich versucht, den laut Einstellung interpretierten Wert auszugeben. Da ein Register immer 2 Byte hat, klappt dies nur bei Interpretation als UInt16 und Int16, erleichtert aber die Analyse von Telegrammen.

1.11.5 Pvi – Korrektur beim Anlegen einer ANSL-Verbindung

Beim manuellen Anlegen einer ANSL-Verbindung wurde der automatisch generierte Verbindungsname mit der falschen Ip-Adresse erstellt.

1.11.6 OpcUaClient/Server – Korrektur bei Eingabe von Datums-Werten

Bei dem seit V5.03 eingesetzten Steuerelement zur Eingabe von Datums-Werten hatte sich ein Fehler eingeschlichen: Bei der Tastatur-Eingabe der Sekunden wurden die Minuten auf den Wert der Stunden gesetzt. Dies ist jetzt korrigiert.

1.11.7 OpcUaClient - Endpoint: Optimierung des Fensters

Der Dialog zum Angeben des Endpoints wurde verbessert. Zum einen kann die Größe des Fensters jetzt geändert werden. Somit ist eine bessere Bedienung auch bei kleinerer Auflösung möglich.

Zum anderen wurde die Parameter-Liste jetzt in die beiden Sektionen „Client“ und „Tool“ eingeteilt. Damit ist eine bessere Zuordnung der Parameter-Funktionalität gegeben.

1.11.8 OpcUaClient – Endpoint - Search: Anpassung der Adapter-Liste

Analog zum Modul „Ethernet“ wurde auch hier die Liste der Ethernet-Adapter optimiert. So wird nun auch die Description des Adapters angezeigt.

1.11.9 OpcUaClient - Browse: Korrektur beim Suchen

Die Suche wurde immer beim Root-Knoten begonnen. Jetzt wird, wie in der Hilfe beschrieben, beim ausgewählten Knoten gestartet. Ist der gesuchte Knoten in diesem Zweig nicht enthalten, wird die Suche beendet.

Außerdem kann die Liste der Suchbegriffe jetzt durch einen Button „...“ verwaltet werden (Ändern der Reihenfolge, Löschen von Einträgen, usw., siehe Punkt 4.).

1.11.10 OpcUaClient – Browsen bei Eingabe einer Nodeld

Bisher musste eine Nodeld immer manuell eingegeben werden. Jetzt kann in vielen Fällen auch ein Dialogfeld (meist über das Kontext-Menü) zum Browsen geöffnet werden.

Implementiert wurde dies in:

- Attributes.Access.Advanced
- Attributes.Access.Direct

- Attributes.Access.Translate
- Subscription – Werte-Fenster
- MethodCall – Nodeld als Eingangs-Argument

1.11.11 OpcUaClient – Speichern und Laden von Werten/Argumenten: Neues Format

Das Format der Dateien zum Speichern von Werten und Argumenten wurde geändert (siehe auch unter OpcUaClient – Hinweise – Speichern und Laden von Werten). Damit werden mehr Datentypen unterstützt. Gespeichert wird nur noch im neuen Format, geladen werden kann auch das alte Format.

Das neue Format besitzt neue Endungen:

Altes Format (Xml-Serialisierung)	Neues Format (Binär-Serialisierung)

Client-Variablen-Wert	
*.mUaClntValue	*.mUaClntValueBin
Client-Argument-Wert	
*.mUaClntArg	*.mUaClntArgBin

1.11.12 OpcUaClient – Optimierung beim Erzeugen von Datentyp-Instanzen

Zum Erzeugen von Instanzen eines bestimmten Datentyps (auch mehrdimensionale Arrays), z.B. beim zufälligen Setzen von Werten oder Erzeugen von Default-Argumenten, werden jetzt dieselben Routinen wie beim OpcUa-Server verwendet. Das erleichtert die Wartung enorm.

1.11.13 OpcUaClient: Unterstützung von Grafik-Datentypen

Die OpcUa-Datentypen für Grafiken (ImageBMP, ImageJPG und ImagePNG), welche als ByteString übertragen werden, werden nun in allen Bereichen durch das neue Dialogfeld (siehe oben) unterstützt (DirectAccess, Subscription, Methoden).

1.11.14 OpcUaClient – Access.Advanced: Erweiterung bei ExtensionObjects

Wenn sich der selektierte Knoten seit dem letzten Lesen nicht geändert hat, wird die Wert-Liste beim erneuten Lesen nicht komplett neu aufgebaut, sondern nur aktualisiert. Dies macht die Anzeige schneller, vor allem, wenn es sich um ein ExtensionObject mit vielen Werten handelt.

Wenn in einem als ExtensionObject übertragenen Wert ein dynamisches Array enthalten ist, konnte folgender Fall auftreten: Hatte sich die Länge des dynamischen Arrays seit dem letzten Lesen verändert, wurde die Liste nicht neu aufgebaut, sondern versucht, die neuen Werte in die alte Liste einzutragen. Da die alte Liste die neuen Werte nicht beinhaltete, kam es zu einer Exception.

Jetzt wird vor dem Füllen der Liste die Anzahl der Bytes des ExtensionObjects geprüft. Unterscheidet sie sich von der Länge beim letzten Lesen, wird die Liste komplett neu aufgebaut. So werden immer alle aktuellen Werte angezeigt.

1.11.15 OpcUaClient – Access.Advanced: Erweiterung für Xml-Elemente

Werte vom Datentyp „XmlElement“ werden nun als String angezeigt und können editiert werden.

Das Speichern/Laden ist leider nicht möglich, da das DotNet-Framework die Serialisierung von Xml-Elementen nicht zulässt (siehe Hinweise „Speichern und Laden von Werten“). Dafür kann der Wert aber als Text-Datei ex- und importiert werden.

1.11.16 OpcUaClient – Access.Advanced: Korrektur für LocalizedTexts

Wenn bei einem LocalizedText eines der Elemente „Locale“ oder „Text“ vom Server nicht besetzt (Null) war, trat eine Exception auf. Jetzt wird stattdessen der Text „Null“ angezeigt.

1.11.17 OpcUaClient – Access.Direct: Korrektur für Strukturen

Wurde ein Struktur-Knoten gelesen, so wurde der gelesene Wert nicht aufgelöst, sondern das Byte-Array des ExtensionObjects angezeigt. Dies ist jetzt korrigiert.

1.11.18 OpcUaClient – Access.Advanced: Optimierung für Strukturelemente

Struktur-Elemente vom Datentyp ‚LocalizedText‘ und ‚QualifiedName‘ werden jetzt besser unterstützt.

1.11.19 OpcUaClient – Access.Advanced: Unterstützung für EnumValueType

Irreguläre Enumerationen (nicht bei 0 beginnend oder nicht durchgehend nummeriert) können durch den Property-Knoten „EnumValues“ beschrieben werden. Dieser Property-Knoten hat dann den OpcUa-Datentyp „EnumValueType“. Er enthält ein Array von Strukturen, bei dem jedes Element ein Element der Enumeration beschreibt. Ein Element enthält also den Enumerations-Wert, einen DisplayName und eine Description. Dieser Datentyp kann jetzt aufgelöst und angezeigt werden.

1.11.20 OpcUaClient – Access.Direct: Erweiterung um indizierten Zugriff

Bei einer Array-Node kann jetzt durch Angabe eines Index-Bereichs auch auf einzelne Array-Elemente (oder einen Bereich davon) lesend und schreibend zugegriffen werden.

1.11.21 OpcUaClient – Access: Erweiterung um Attribut-Zugriff

Es gibt jetzt einen neuen Reiter „Attributes“, welcher das Schreiben auf schreibbare Attribute ermöglicht.

1.11.22 OpcUaClient - Subscription: Korrektur beim Hinzufügen von Event-Items

Um festzustellen, ob es sich bei einem Knoten um ein Event-Item handelt, wird das bitcodierte Attribut „EventNotifier“ verwendet. Beim Hinzufügen wurde der Wert nicht bitcodiert geprüft, so dass ein Item nicht hinzugefügt wurde, wenn ausser dem Bit „SubscribeToEvents“ auch noch andere Bits gesetzt waren. Dies ist jetzt korrigiert.

1.11.23 OpcUaClient - Subscription: Neue Funktion Archive-Mode

Es kann jetzt ein Archive-Mode ein- und ausgeschaltet werden. Bei aktivem Modus werden die Werte nicht mehr aktualisiert. So ist ein „Screenshot“ der Werte möglich. Eingaben werden nicht auf den Server übertragen, sondern lokal gehalten. Außerdem können Werte einzeln oder als gesamte Liste am Server gelesen und geschrieben werden. Da die Werte in der Konfiguration des OpcUa-Monitors jetzt mitgespeichert und geladen werden, können so auch ‚Rezepte‘ verwendet werden. Z.B. kann eine Parameter-Liste aus einer Konfiguration geladen und übertragen werden.

1.11.24 OpcUaClient – MethodCall: Optimierung bei Ausgabe

Wenn der Rückgabe-Status eines Aufrufs ungleich „Good“ ist, wird die Ausgabe-Zeile rot hinterlegt.

1.11.25 OpcUaClient – MethodCall: Korrektur bei dynamischen Arrays

Ist ein Argument ein dynamisches Struktur-Array, so konnten zwar Elemente hinzugefügt, aber nicht mehr entfernt werden. Dies ist behoben.

1.11.26 OpcUaClient – MethodCall: Korrektur für B&R-Authorisierungs-Methoden

Beim Hinzufügen einer der beiden Methoden ‚GetAuthorizations‘ oder ‚SetAuthorizations‘ werden die Eingangs-Argumente schon vorbefüllt, damit die darin enthaltenen dynamischen Arrays (Nodes[], Roles[] und Rights[]) nicht manuell erweitert werden müssen.

Beim mehrmaligen Hinzufügen einer dieser Methoden wurde eine Exception ausgelöst. Dies ist jetzt behoben.

1.11.27 OpcUaClient – MethodCall: Erweiterung für Xml-Elemente

Argumente vom Datentyp „XmlElement“ (i=16) werden nun als String angezeigt und können editiert werden.

Das Speichern/Laden ist leider nicht möglich, da das DotNet-Framework die Serialisierung von Xml-Elementen nicht zulässt (siehe Hinweise „Speichern und Laden von Werten“). Dafür kann der Wert aber als Text-Datei ex- und importiert werden.

1.11.28 OpcUaClient – MethodCall: Erweiterung für Datentyp ‚TimeZoneDataType‘

Argumente vom Datentyp „Opc.Ua.TimeZoneDataType“ (i=8912) werden nun als String angezeigt und können editiert werden.

1.12 RnCommTest V5.03

1.12.1 Korrektur Zertifikats-Anzeige

Bei OpcUa-Zertifikaten werden sogenannte Extensions verwendet. Bei der Anzeige eines Zertifikats wurde bei manchen Erweiterungen nicht der korrekte Wert angezeigt. Dies wurde überarbeitet.

1.12.2 OpcUaClient: Optimierung der Anzeige und Eingabe von DateTime

Der OpcUa-Datentyp ‚DateTime‘ unterstützt einen Bereich zwischen 1601-01-01 00:00:00 bis 9999-12-31 23:59:59 mit einer Auflösung von 100 Nanosekunden. Bisher wurde zur Anzeige/Eingabe das Steuerelement ‚DateTimePicker‘ vom Windows-Framework verwendet. Dieses unterstützt aber nur einen Bereich von 1753-01-01 00:00:00 bis 9998-12-31 23:59:59. War ein anzuzeigender Zeitpunkt ausserhalb dieser Anzeige-Grenzen, so wurde die aktuelle Uhrzeit angezeigt und folgende Warnung ausgegeben:

Value is out of range of the display control. The display is set to the current time. The real value = 01.01.1601 00:00:00

Nun wird ein selbstgeschriebenes Steuerelement verwendet, dass den ganzen OpcUa-Bereich unterstützt, so dass alle Werte korrekt angezeigt werden können.

Bei der Ausgabe von DateTime-Werten werden jetzt auch die OpcUa-Ticks (100 Nanosekunden-Ticks) in Klammern angezeigt. In der Advanced-Liste werden sie in der letzten Spalte ‚Description‘ angezeigt.

Außerdem wurde neue Informationen in das Kapitel „OpcUaClient – Hinweise – Besonderheiten der Datentypen - DateTime“ aufgenommen.

1.12.3 OpcUaClient – Server-Properties: Änderung des Benutzers

Bei einer bestehenden Verbindung können jetzt sowohl der Benutzer als auch die bevorzugten Sprachen ohne Disconnect geändert werden.

1.12.4 OpcUaClient - Browse: Korrektur des Auswahl-Verhaltens

Wenn ein Knoten selektiert und dann gleich die Maus verschoben wurde, wurde zwischenzeitlich der Knoten unter der verschobenen Maus selektiert. Das löste eine Auffrischung des zwischenzeitlich selektierten Knotens aus und wirkte sich daher zeitlich negativ aus, wenn der Knoten sehr viele Unterknoten hatte. Dieses Verhalten wurde korrigiert.

1.12.5 OpcUaClient - Attributes: Erweiterung bei Value-Anzeige

Über das Kontextmenü kann jetzt die Anzeige des Attributs ‚Value‘ zwischen Dezimal (Standard), Hexadezimal und Binär umgeschaltet werden.

1.12.6 OpcUaClient - Access: Erweiterung bei ExtensionObjects

Eine Nodell, welche als Element einer Struktur in einem ExtensionObject übertragen wird, wird nach bestimmten Regeln in das zu übertragende Byte-Array gepackt, um Platz zu sparen. So wird z.B. unterschieden, ob die Nodell in 2, 4 oder mehr Bytes Platz findet.

Bis jetzt war die Unterstützung für 2-Byte-Nodells nicht implementiert, so dass in diesem Fall die Warnung „Cannot solve byte array as structure typ: Ex=Nodell with Format ‚NumericTwoBytes‘ is not supported!“ ausgegeben wurde.

Dieses Format wird jetzt unterstützt.

1.12.7 OpcUaClient - Subscription: Optimierung bei Event-Notification

Wird bei einem EventItem die Notification ausgegeben, werden im aufklappbaren Teil die Felder-Werte dargestellt. Ist diese Darstellung jedoch zu lang, wird sie gekürzt. Mit einem Doppelklick auf die Zeile öffnet sich ab V5.02 ein Dialogfeld, mit dem die ungekürzten Werte eingesehen werden können.

Allerdings wurden hier benutzerdefinierte Strukturen, welche als ExtensionObject (Byte-Array) gesendet wurden, nicht in die ursprüngliche Struktur aufgelöst. Dies ist jetzt der Fall.

1.12.8 OpcUaClient - Performance: Erweiterung um Action-Modus

Bei OpcUa kann eine ganze Liste von Knoten als ein Aufruf gelesen/geschrieben werden. Bis jetzt wurden bei diesem Test-Modul alle Knoten immer als Liste gelesen/geschrieben.

Es kann jetzt festgelegt werden, ob die Aktion (Read/Write) im Single- oder List-Modus ausgeführt wird.

1.12.9 OpcUaServer: Optimierung der Anzeige und Eingabe von DateTime

Analog zum OpcUa-Client (siehe oben) wurde auch hier das selbstgeschriebene Steuerelement verwendet, das den ganzen OpcUa-Bereich unterstützt, so dass alle Werte korrekt angezeigt werden können. OpcUa unterstützt einen Bereich von 1601-01-01 00:00:00:000ms.000µs:000ns bis 9999-12-31 23:59:59:999ms:999µs:900ns. Alle Grenzwerte im Tool wurden darauf angepasst.

1.12.10 OpcUaServer – Settings: Neue Timer

Die Anzahl der verwendeten Timer wurde von bisher 4 auf nun 8 erweitert.

1.12.11 OpcUaServer - AddressSpace: Erweiterung um Struktur-Arrays

Im Baum können jetzt Struktur-Array-Knoten erzeugt werden. Dazu wird zur Vorlage eine Struktur gewählt und dann das Array mit Dimensionen und deren Länge angegeben.

1.12.12 OpcUaServer - AddressSpace: Erweiterung um Methoden

Im Baum können jetzt Methoden-Knoten gesetzt und parametrisiert werden. Durch einen einfachen Code-Interpreter können die Ausgangs-Argumente bei jedem Aufruf besetzt werden.

1.12.13 OpcUaServer - AddressSpace: Schnelles Setzen von Knoten-Parametern

Beim Setzen von Parametern durch das Kontext-Menü ‚Set node parameter to all sub nodes...‘ können jetzt bestimmte Parameter schnell und bequem an allen Unterknoten gesetzt werden.

1.12.14 OpcUaServer - AddressSpace: Korrektur beim Setzen von Source-Settings

Durch das Kontext-Menü ‚Set source settings to all sub nodes...‘ werden jetzt auch die Element-Knoten von Arrays gesetzt.

1.12.15 OpcUaServer – Events: Mapping von Feld-Werten

Bisher musste der Wert eines Feldes fest parametrisiert werden. Jetzt kann bei jedem Feuern auch der Wert eines beliebigen Variablen-Knotens übernommen werden, dessen Datentyp zum Feld passt. Ist der Datentyp des Felds ‚BaseDataType‘, so können auch Struktur-Knoten ausgewählt werden.

1.12.16 OpcUaServer – Events: Zyklisches Feuern

Pro Instanz kann jetzt ein zyklisches Feuern über einen Timer parametrisiert werden. Das zyklische Feuern von Events wird nicht im Haupt-Ausgabe-Fenster, sondern auf dem Reiter ‚Diag‘ ausgegeben. Diese Ausgabe kann unter ‚Settings‘ parametrisiert werden. Manuell gefeuerte Events werden zusätzlich immer auch im Haupt-Ausgabe-Fenster ausgegeben.

1.12.17 OpcUaServer – Diag: Überarbeitung der Ausgaben bei MonitoredItems

Die Ausgaben bei Aktionen an MonitoredItems (Create, Modify, Delete) sind jetzt detaillierter. Außerdem wurden bis jetzt nur Ausgaben für MonitoredItems gemacht, deren Knoten sich im benutzerdefinierten Adressraum befinden. Jetzt kann über die Settings eingestellt werden, daß diese Ausgaben auch für Core-Nodes (also Knoten, die nicht benutzerdefiniert sind) gemacht werden.

1.12.18 OpcUaServer – Beispiel-Konfiguration: Neue Funktionen beispielhaft integriert

Die Beispiel-Konfiguration ‚Example_General‘ wurde erweitert, um die neuen Funktionen (z.B. Struktur-Arrays) zu zeigen.

1.13 RnCommTest V5.02

1.13.1 OpcUa – Stack

Als Basis für den OpcUa-Client/-Server wird der OpcUa-Stack der Opc-Foundation verwendet. Um bestimmte Funktionalitäten des Tools umzusetzen, wird dieser Stack an manchen Stellen verändert/erweitert. Wegen einem Fehler beim Backup des Tools ging die zuletzt verwendete Version des Stack 1.03.342 verloren. Deshalb wurde die Original-Version des Stacks eingesetzt und die Änderungen seit RnCommTest V5.00 so gut wie möglich nachgeführt. Allerdings könnte es sein, dass evtl. Änderungen dadurch verloren gingen und das Tool sich jetzt anders verhält.

1.13.2 OpcUaClient: Status auch als Hex-Zahl

Jeder OpcUa-Status wird jetzt nicht mehr nur als englischer Klartext ausgegeben („BadUserAccessDenied“), sondern auch als Hex-Zahl („0x801F0000= BadUserAccessDenied“). So können Fehler-Status besser analysiert werden.

1.13.3 OpcUaClient – Attributes: Korrektur beim Ermitteln der Browse-Pfade

Wenn bei einem Knoten über den Kontextmenüpunkt ‚Get Absolute BrowsePathes‘ die Browse-Pfade ermittelt wurden und dieser Knoten mehr als 9 Pfade hatte, trat eine Exception auf. Jetzt bricht diese Funktion bei mehr als 99 Pfaden ab. Der letzte Eintrag enthält dann den Hinweis ‚Max count of pathes to determine reached!‘.

1.13.4 OpcUaClient - Access: Erweiterung bei Enumerationen

Bei manchen Servern, wie z.B. beim B&R-Server ab AR 4.6x, wird eine Enumerations-Variable nicht als Datentyp ‚Int32‘ ausgeliefert, sondern mit dem Datentyp der Enumeration. Wurde eine Struktur gelesen oder geschrieben, die so eine Enumeration enthält, konnte sie nicht aufgelöst werden. Dies wurde jetzt erweitert.

1.13.5 OpcUaClient - Subscription: Erweiterung

Ab jetzt kann ein Knoten auch mehrmals hinzugefügt, jedoch für jeden Eintrag eine eigene Parametrierung vorgenommen werden. So kann z.B. ein VariableItem mit verschiedenen Sampling-Intervallen/Filtern oder ein EventItem mit verschiedenen Filtern/Feldern abonniert werden. Damit kann ein Server besser getestet und die Funktionalität einer Subscription besser veranschaulicht werden.

1.13.6 OpcUaClient - Subscription: Erweiterung bei Event-Notification

Wird bei einem EventItem die Notification ausgegeben, werden im aufklappbaren Teil die Felder-Werte dargestellt. Ist diese Darstellung jedoch zu lang, wird sie gekürzt. Mit einem Doppelklick auf die Zeile öffnet sich jetzt ein Dialogfeld, mit dem die ungekürzten Werte eingesehen werden können.

1.13.7 OpcUaClient - Subscription: Änderung/Erweiterung bei EventFilter

Beim Abonnieren von Events kann die Liste der Zusatzinformationen (Fields) als Filter parametrierbar werden.

Bis jetzt waren nur ein paar Felder automatisch in der Liste. Jetzt werden automatisch alle Felder aller Events abonniert, aber nur die mit einem Wert ungleich ‚Null‘ ausgegeben. So ist das Analysieren von Events deutlich leichter.

Außerdem können Felder des BaseEventTypes, welches jedes Event liefert, nicht mehr gelöscht werden. Zur besseren Unterscheidung zwischen Event-Typen und -Felder werden jetzt Icons in den Listen angezeigt. Zusätzlich ist jetzt auch möglich, nicht nur Felder hinzuzufügen, welche auf der 1. Ebene des Event-Typs sind, sondern auch weiter untenliegende (Properties).

Außerdem wurden Buttons implementiert, mit denen sich alle Felder aller Events oder die Felder des selektierten Event-Typs hinzufügen sowie die Liste in den Grundzustand versetzen lassen.

1.13.8 OpcUaClient – Subscription: Korrektur bei Arrays

Bei Abonnieren von Array-Knoten konnte unter gewissen Umständen ein Fehler auftreten, wenn die Option ‚Split Arrays‘ angehakt war, die Element-Knoten aber nicht freigegeben waren.

1.13.9 OpcUaClient – Access.Advanced: Erweiterung bei Enum

Eine irreguläre Enumeration wurde als Element einer Struktur nicht aufgelöst, sondern als Int32 angezeigt.

1.13.10 OpcUaClient – Access.Advanced: Korrektur bei Arrays

Das Auffrischen von Array-Werten wurde geändert. Wenn der selektierte Knoten nicht geändert, sondern nur ein erneuter Read ausgeführt wird, wird die Liste normalerweise nicht mehr neu erzeugt, sondern lediglich die Spalte ‚Value‘ gesetzt. Dies optimiert das Auffrischen der Anzeige. Ein dynamisches Array kann aber bei jedem Read die Anzahl seiner Elemente ändern. Das führte zu einem Fehler, wenn das neue Array mehr Elemente hatte als das alte, weil zu wenig Listeneinträge vorhanden waren. Jetzt wird bei erneutem Lesen die Anzahl der Elemente geprüft. Hat sie sich geändert, so wird die Liste mit der aktuellen Anzahl von Elementen neu gefüllt.

1.13.11 OpcUaClient – Access.Advanced: Korrektur bei ByteString-Wert

Wurde ein Knoten vom Datentyp ‚ByteString‘ angewählt, so konnte zwar der enthaltene Text mit dem Dialog ‚Show as text‘ angezeigt und auch verändert werden, die Länge des Strings war aber auf die ursprüngliche Länge begrenzt. Jetzt kann auch ein längerer Text eingegeben werden.

1.13.12 OpcUaClient – MethodCall: Korrektur bei ByteString-Argument

Wenn ein Argument vom Datentyp ‚ByteString‘ war, wurde es nicht korrekt angezeigt und konnte auch nicht editiert werden. Jetzt kann durch einen Doppelklick der Dialog zum Anzeigen/Editieren geöffnet werden. In diesem kann auch der Dialog ‚Show as text‘ benutzt werden.

1.13.13 OpcUaClient – MethodCall: Korrektur bei ExtensionObject-Array

Wenn ein Argument ein ExtensionObject-Array war, wurde bei einem Doppelklick nur das erste Element angezeigt.

1.13.14 OpcUaClient – Access.Advanced: Importieren einer Datei

Ist ein Wert vom Datentyp ‚String‘, ‚Byte()‘ oder ‚ByteString‘, so kann jetzt über den neuen Kontext-Menüpunkt ‚Import file‘ eine beliebige Datei geladen werden. Dies ist auch bei Methoden-Argumenten möglich. Besonders nützlich ist dies bei Text-Dateien oder zum Laden eines Zertifikats als ByteString.

1.13.15 OpcUaClient – Access.Advanced: Exportieren als Datei

Ist ein Wert vom Datentyp ‚String‘, ‚Byte()‘ oder ‚ByteString‘ so kann jetzt über den neuen Kontext-Menüpunkt ‚Export file‘ eine Datei mit beliebiger Endung gespeichert werden. Dies ist auch bei Methoden-Argumenten möglich. Besonders nützlich ist dies bei Text-Dateien oder zum Speichern eines Zertifikats als ByteString.

1.13.16 OpcUaServer - AddressSpace: Änderung des Layouts

Zur Vorbereitung neuer Funktionalitäten wurde das Layout dieser Ansicht geändert. Zum einen sind die verschiedenen Anzeigen anders aufgeteilt. Zum anderen werden diese Anzeigen aufgrund der NodeClass des selektierten Knotens ein- und ausgeblendet.

1.13.17 OpcUaServer - AddressSpace: Hervorhebung von benutzerdefinierten Knoten

Da jetzt für neue Funktionalitäten der Baum auch teilweise mit den Knoten der Opc-Foundation gefüllt wird, werden benutzerdefinierte Knoten jetzt **fett** dargestellt.

1.13.18 OpcUaServer - AddressSpace: Erweiterung um Event-Typen

Im Baum werden jetzt Event-Typen samt deren Felder angezeigt. Dadurch ist es möglich, eigene Event-Typen zu definieren.

1.13.19 OpcUaServer: Neue Seite „Events“

Auf dieser Seite können Events instanziiert, parametrisiert und gefeuert werden.

1.14 RnCommTest V5.01

1.14.1 Tool: Bessere Strukturierung durch Unterordner für Daten-Dateien

Jedes Modul kann evtl. seine spezifischen Einstellungen in einer oder mehreren Dateien speichern. Dies geschah bei den meisten im Installationsverzeichnis. Da die Module immer mehr werden, wurden auch die Daten-Dateien immer mehr. Um die ‚Unordnung‘ im Installationsverzeichnis zu beheben, sind jetzt die Dateien in mehreren Unterordnern strukturiert:

Modul	Unterordner
Rs232/Udp	./Frames/
TcpClient/Server	./Frames/
TcpTextClient	./TextFrames/
ModbusTcpMaster	./Modbus/MbGroups/
ModbusTcpSlave	./Modbus/MbAddresses/
PviWatch	./PviWatch/
OpcDaClient	./OpcDaConfigs/
OpcUaClient	./OpcUa/Client/
OpcUaServer	./OpcUa/Server/

1.14.2 Tool: Korrektur beim Laden-/Speichern-Dialog

Bei allen Modulen, bei denen Daten geladen oder gespeichert werden, wird jetzt beim Öffnen des Lade-/Speichern-Dialogs in das Verzeichnis gesprungen, in dem die letzte Datei geladen oder gespeichert wurde.

1.14.3 OpcUaClient: Neue Einstellungen zur Auflösung von Zeitangaben

OpcUa unterstützt sehr hochauflösende Zeitangaben. Daher kann jetzt angegeben werden, in welcher Auflösung jeweils Zeitstempel und DateTime-Werte ausgegeben werden.

Angabe	Bisher	Jetzt	Standard
Zeitstempel	[ms]	[ms], [µs], [ns] oder [ps]	[ms]
DateTime-Wert	[s]	[s], [ms], [µs] oder [ns]	[s]

Dabei entsprechen:

[s]	Sekunden
[ms]	Milisekunden
[µs]	Mikrosekunden
[ns]	Nanosekunden
[ps]	Pikosekunden

1.14.4 OpcUaClient – Attributes: Erweiterung zum Ermitteln von Browse-Pfaden

Im Kontextmenü der Attribut-Liste gibt es jetzt die neue Funktion ‚Get Absolute BrowsePath from Tree‘. Mit diesem kann der BrowsePath auch dann ermittelt werden, wenn der Server fehlerhafterweise keine Rückwärts-Referenzen gesetzt hat.

1.14.5 OpcUaClient – Access: Korrektur bei Strukturen mit Int64, UInt64 und Localized-Text

Wenn ein ExtensionObject eine Struktur mit Elementen vom Datentyp ‚Int64‘, ‚UInt64‘ oder ‚LocalizedText‘ enthält, konnte die Struktur nicht korrekt aufgelöst werden. Da diese Datentypen bei einer SPS nicht vorkommen, fiel dies erst jetzt auf und wurde korrigiert.

1.14.6 OpcUaClient – New Endpoint: Korrektur beim Laden

Wenn eine Konfiguration geladen wurde, der enthaltene Benutzername oder das Passwort aber noch nicht in der Auswahl war, wurden sie nicht übernommen.

1.14.7 OpcUaClient – OpcUa Monitor: Korrektur

Der Monitor funktionierte in den letzten Versionen vermutlich aufgrund neuer Einstellungen nicht mehr. Er wurde wieder funktionsfähig gemacht.

Außerdem wurden analog zum Client die Ordner und Datei-Endungen für die Konfigurationen geändert. Folgende Monitor-Dateien wurden in neue Ordner verschoben:

Vorher	Nachher
-----	-----
./OpcUaConfigs/xxx	./OpcUa/Client/Configs/xxx

Sollten schon vorhandene Dateien einer vorigen Version verwendet werden müssen, können diese in der neuen Version in die neuen Ordner kopiert werden.

Außerdem wurden die Endungen für die Monitor-Konfigurations-Dateien geändert.

Vorher

Nachher

*.ouc

*.mUaClntMonCfg

Da sich das Format der Inhalte nicht geändert hat, können auch noch die alten Endungen geladen werden. Gespeichert wird nur noch mit der neuen Endung.

1.14.8 OpcUaServer – Diag: Korrektur

Beim Stoppen des Servers werden jetzt alle Listen gelöscht.

1.14.9 OpcUaServer – Diag: Korrektur

Einige Subscription-Werte (z.B. „MonitoredItemCount“) wurden nicht korrekt aktualisiert.

1.14.10 OpcUaServer: Neue Einstellung zur Auflösung von Zeitangaben

OpcUa unterstützt sehr hochauflösende Zeitangaben. Daher kann jetzt angegeben werden, in welcher Auflösung DateTime-Werte eingegeben werden.

Angabe	Bisher	Jetzt	Standard
DateTime-Wert	[s]	[s], [ms], [µs] oder [ns]	[s]

Dabei entsprechen:

[s]	Sekunden
[ms]	Milisekunden
[µs]	Mikrosekunden
[ns]	Nanosekunden

1.14.11 OpcUaServer – AdressSpace - Source: Korrektur des Operanden für Datentyp ‚DateTime‘

Ist für eine Variable mit dem Datentyp ‚DateTime‘ eine Action (z.B. ‚Increment‘) eingestellt, so war der Operand bisher vom Datentyp ‚Timespan‘, mit dem sich eine Zeitspanne interpretieren lässt. Dieser hat aber nicht die Auflösung, die neuerdings benötigt wird. Außerdem funktionierten andere Zeitspannen als 1 Sekunde nicht korrekt.

Deshalb wurde der Datentyp des Operanden auf ‚Long‘ geändert und stellt dadurch die Anzahl der Ticks bereit. Ein Tick bedeutet 100 Nanosekunden.

Beim Laden von alten Konfigurations-Dateien wird der Wert automatisch angepasst.

1.14.12 OpcUaServer – AdressSpace - Source: Neue Action für Datentype ‚DateTime‘

Für eine Variable mit dem Datentyp ‚DateTime‘ kann jetzt eine Action ‚SetCurrent‘ gesetzt werden. Sie besetzt den Wert mit der aktuellen Zeit.

1.14.13 OpcUaServer – AdressSpace: Unterstützung von Strukturen

Knoten können jetzt als ‚UserDatatype‘ parametrisiert werden. Dadurch wird der Knoten zu einer Struktur mit den unterliegenden Variablen-Knoten als Struktur-Elemente. Die Deklarations-Informationen der Struktur werden automatisch beim Starten erzeugt.

So kann ein Struktur-Knoten als ExtensionObject gelesen und geschrieben werden. Dies ist auch geschachtelt möglich.

1.14.14 OpcUaServer – AdressSpace: Änderbare Reihenfolge der Knoten

Die Reihenfolge der Unterknoten innerhalb eines Knotens ist normalerweise unerheblich. Nur wenn der Oberknoten wie neuerdings möglich eine Struktur darstellt, wird die Reihenfolge dazu benutzt, die benötigten Struktur-Informationen zu erzeugen. Das bedeutet, der Wert des Oberknotens wird in dieser Reihenfolge als ExtensionObject zusammengepackt.

Deshalb kann jetzt die Reihenfolge von Unterknoten bestimmt werden.

1.14.15 OpcUaServer – AdressSpace: Umwandlung von Object-Knoten in Struktur-Knoten und umgekehrt

Bei vorhandenen Adressräumen kann ein Knoten vom Typ ‚Object‘, welcher Variablen-Knoten enthält und bisher zur Strukturierung der Daten verwendet wurde, zu einem Struktur-Knoten gewandelt werden und umgekehrt.

1.14.16 OpcUaServer: Angabe der unterstützten Sprachen

Der Knoten ‚Root/Objects/Server/ServerCapabilities/LocaleIdArray‘ enthält jetzt die vom Benutzer angelegten Sprach-Kürzel.

1.14.17 ModBusTcpMaster: Erweiterung um Double

Es können jetzt auch 4 hintereinander folgende Register als Double64 (= 8 Byte) interpretiert werden. Das entspricht dem SPS-Typen ‚LREAL‘.

1.14.18 ModBusTcpMaster: Erweiterung um Swapping

Ein Register besitzt 2 Byte. Hintereinander folgende Register können als ein Wert mit mehr als 2 Byte interpretiert werden (z.B. Uint32 mit 4 Byte = 2 Register).

Manche Slaves liefern diese Bytes aber in vertauschter Reihenfolge aus (Endian-Format), z.B. neuere B&R-SPS mit Intel-Prozessor.

Jetzt kann bei Interpretation als Uint32, Int32, Float32 und Double64 das Swapping aktiviert werden. Damit werden diese Werte korrekt interpretiert.

1.14.19 ModBusTcpSlave: Erweiterung um Interpretation und Swapping

Wenn eine Adresse aus mehreren Registern besteht, kann es jetzt auch als Datentyp interpretiert werden, dessen Wert in einer zusätzlichen Spalte ausgegeben wird.

Über ein Swapping kann die Reihenfolge der Bytes zur Interpretation als Uint32, Int32, Float32 und Double64 vertauscht werden.

1.14.20 Pvi – Umstieg auf neue Version

Bei der benutzten Bibliothek ‚PviServices.dll‘ wurde von der Pvi-Version V3.0.2.3014 auf V4.5.2.57 umgestellt, welche beim AS4.5 mitinstalliert wird.

1.14.21 Pvi – Erweiterung um DeviceType ‚ANSLTcp‘

Seit geraumer Zeit (ab AR V4.08) gibt es die PVI-Linie ‚ANSLTcp‘, welche ähnlich funktioniert wie ‚IN-A2000‘ (DeviceType ‚TcpIp‘), jedoch schneller ist. Mittlerweile wird vom Automation Studio zur Online-Verbindung nur noch ANSL verwendet.

Aus diesem Grund wurde der neue DeviceType ‚ANSLTcp‘ implementiert. Er sollte ab jetzt statt ‚TcpIp‘ verwendet werden.

1.14.22 Pvi – Korrektur: DeviceType ‚AR000‘ entfernt

Der DeviceType ‚AR000‘ ist mittlerweile obsolet und wurde daher entfernt. Er kann mit ‚ANSLTcp‘ abgedeckt werden.

1.14.23 Pvi – Logger: Korrektur

Die Anzeige eines Logbuchs wurde auf die neue ‚PviServices.dll‘ angepasst. Außerdem wird jetzt zu jedem Logeintrag auch die EventId angezeigt.

1.15 RnCommTest V5.00

1.15.1 Dialog „ShowText“: Erweiterung um Json-Format

Das Dialogfeld wird in manchen Modulen verwendet (z.B. OpcUa-Client), um lange Texte anzuzeigen. Texte können jetzt auch im Json-Format angezeigt und somit die Lesbarkeit deutlich verbessert werden.

1.15.2 Umstellung des OpcUa-Stacks

Es wurde von OpcUa-Stack V1.03.341 auf V1.03.342 umgestellt.

1.15.3 Neues Modul OpcUaServer

Es wurde ein OpcUa-Server implementiert, an dem ein benutzerdefinierter Adressraum parametrisiert sowie sehr viele Einstellungen gesetzt werden können.

1.15.4 Neues Kapitel ‚Translations‘

Da für den OpcUa-Server übersetzbare Texte benötigt werden, wurde eine Verwaltung solcher Texte implementiert. Da diese evtl. später auch für andere Module benutzt werden wird, wurde diese in einem eigenen Hauptkapitel ‚Translations‘ beschrieben.

1.15.5 OpcUa – Änderung der Ordner-Struktur und der Datei-Endungen

Da die benötigten Dateien für OpcUa (Client und Server) immer mehr werden, wurde zur besseren Verteilung die Ordner-Struktur innerhalb des Tool-Verzeichnisses geändert. Folgende Client-Dateien wurden in neue Ordner verschoben:

Vorher	Nachher
-----	-----
./OpcUaCertificates\Client\xxx	./OpcUa/Client/Certificates\xxx
./RnCommTestOpcUaClientConfig.xml	./OpcUa/Client/AppConfigClient.xml
./OpcUaConfigs\xxx	./OpcUa/Client/Configs\xxx
./OpcUaValues\xxx	./OpcUa/Client/Values\xxx

Die Dateien für den Server wurden analog dazu in ‚./OpcUa/Server/‘ gelegt.

Sollten schon vorhandene Dateien einer vorigen Version verwendet werden müssen, können diese in der neuen Version in die neuen Ordner kopiert werden.

Außerdem wurden die Endungen für die Client-Dateien geändert.

Vorher	Nachher
-----	-----
Client-Config	
*.mous	*.mUaClntCfg
Client-Variablen-Wert	
*.rox	*.mUaClntValue
Client-Argument-Wert	
*.roa	*.mUaClntArg

Da sich das Format der Inhalte nicht geändert hat, können auch noch die alten Endungen geladen werden. Gespeichert wird nur noch mit der neuen Endung.

1.15.6 OpcUaClient – Verbindung: Korrektur

Die Einstellung ‚Replace discovered hostname with IP‘ wurde ab einer der letzten Versionen nicht mehr berücksichtigt. Das ist jetzt wieder der Fall.

1.15.7 OpcUaClient – Browse: Optimierung

Der von der Opc-Foundation spezifizierte Knoten ‚Server‘ wird bei der Sortierung immer am Ende der Liste angezeigt.

1.15.8 OpcUaClient – Attributes: Erweiterung

Über den Kontextmenüpunkt ‚Show as text‘ werden jetzt auch XML-Elemente korrekt angezeigt wie schon in AccessAdvanced.

1.15.9 OpcUaClient – Access.Advanced: Korrektur

Jetzt werden auch UnitId's angezeigt, die einen nur zweistelligen CommonCode haben.

1.15.10 OpcUaClient – Access.Advanced: Korrektur

Bei String-Arrays wurde in der Value-Spalte der Index mit angezeigt, z.B. „[0]=xxx“. Das ist jetzt korrigiert.

1.15.11 OpcUaClient – Access.Advanced: Optimierung

Bei Enum-Arrays werden jetzt (wie bei einer skalaren Enum) Auswahlfelder mit den gültigen Texten angezeigt.

1.15.12 OpcUaClient – Access.Advanced: Optimierung

Die Anzeige eines LocalizedTexts und eines Arrays davon wurde optimiert, so dass sie jetzt auch editierbar sind.

1.15.13 OpcUaClient – Subscription: Korrektur

Beim Kontextmenüpunkt ‚Item Settings For All‘ können die Parameter ‚Filter Trigger‘, ‚Filter Type‘ und ‚Filter Value‘ eingestellt werden. Sie wurden jedoch beim Hinzufügen eines neuen Items nicht berücksichtigt. Jetzt werden neue Items auch mit diesen Werten vorbesetzt.

1.15.14 OpcUaClient – Subscription: Korrektur

Unter Umständen wurde ein multidimensionales Array nicht gesplittet, wenn dies eingestellt war. Dies ist jetzt der Fall.

1.15.15 OpcUaClient – Subscription: Optimierung

Beim Hinzufügen eines Arrays, welches in seine Elemente gesplittet werden soll, werden nun Property-Knoten (z.B. ‚EngineeringUnit‘) nicht mehr hinzugefügt.

1.15.16 OpcUaClient – Subscription: Korrektur

Wenn das Splitting von Arrays eingestellt war, der Array-Knoten aber keine Elemente veröffentlicht, wurden natürlich kein Unter-Knoten hinzugefügt. Jetzt wird stattdessen der Haupt-Knoten hinzugefügt.

1.15.17 OpcUaClient – ServerDiag: Erweiterung

Es können jetzt auch die Werte der SecurityDiagnostics eines verbundenen Clients sowie ein eventuell vorhandenes Client-Zertifikat angezeigt werden.

1.16 RnCommTest V4.07

1.16.1 Angabe einer Entwicklungs-Version

Wird eine Version verwendet, die sich noch in der Entwicklung befindet, wird in der Titelzeile zusätzlich der Text „Development Version“ angezeigt. Bei einer Release-Version fehlt dieser Text.

1.16.2 Korrektur der Wert-Eingabe

Bei Wert-Eingaben vom Datentyp Single und Double können nicht nur Ziffern, sondern alle Zeichen eingegeben werden. Eine aus der unmöglichen Konvertierung entstehende Exception wird jetzt abgefangen und der vorherige Wert bleibt erhalten. Dies gilt für alle Werte-Tabellen und –Listen.

1.16.3 Dialog „ShowText“: Optimierung und Erweiterung

Das Dialogfeld wird in manchen Modulen verwendet (z.B. OpcUa-Client), um lange Texte anzuzeigen. Es ist jetzt auf AutoSize gestellt, um auch bei bestimmten Anzeige-Einstellungen korrekt dargestellt zu werden.

Außerdem gibt es jetzt die Möglichkeit, nach enthaltenem Text zu suchen.

1.16.4 Neues Dialogfeld zum Verwalten von Listen

Text-Listen, welche in manchen Auswahlfeldern zur Auswahl stehen, können nun in einem neuen Dialogfeld verwaltet werden (Ändern der Reihenfolge, Löschen von Einträgen, usw.; siehe Punkt 4.). Es wurde bei den meisten Auswahlfeldern im ganzen Tool implementiert, erkennbar durch einen rechts vom Auswahlfeld befindlichen Button „...“.

1.16.5 OpcUaClient – Endpoint-New: Speichern und Laden der Einstellungen

Zum schnellen Wechseln der Einstellungen aller 3 Tabs (New, Discover und Search) können diese jetzt explizit gespeichert und geladen werden.

1.16.6 OpcUa-Client – Browsing

Bei verschiedenen Browse-Aktionen wird jetzt die Sanduhr eingeblendet. Das ist hilfreich, wenn der verbundene Server langsam reagiert (z.B. bei Aggregats-Servern, die sich ihrerseits einen Teil des Adressraums von anderen Servern holen müssen).

1.16.7 OpcUa-Client – Attribut-Liste: Erweiterung „Show as Text“

Im Kontext-Menü gibt es einen neuen Punkt „Show as Text“. Er funktioniert so wie im Fenster „Advanced“ und zeigt den Wert des angewählten Attributs als Text. Dies ist besonders nützlich beim Attribut „Value“, wenn es sich dabei um einen langen String, einen Byte-String oder ein Byte-Array handelt, das als Text interpretiert werden kann (z.B. bei einem Data-Dictionary).

1.16.8 OpcUaClient – Access.Advanced: Optimierung bei der Anzeige von Arrays

Beim Anzeigen eines 1-dimensionalen Arrays eines Standard-Datentypen wurde die Geschwindigkeit zum Aufbauen der Liste signifikant verbessert. Außerdem kann der Aufbau bei besonders langen Listen durch einen Cancel-Button abgebrochen werden.

1.16.9 OpcUaClient – Access.Direct: Auswahl der Nodetd

Für die Nodetd gibt es jetzt ein Auswahl-Feld, in denen die letzten Eingaben gespeichert werden.

1.16.10 OpcUaClient – HistoricalAccess: Korrektur

Irrtümlicherweise wurde im Kontext-Menü des großen Trends der Punkt „ShowLimits“ angezeigt. Das ist behoben.

1.16.11 OpcUaClient – ServerDiag: Erweiterung

Dieses Modul wurde um Ausgabefelder erweitert, welche mitloggen, wann eine Session bzw. Subscription hinzugefügt oder gelöscht wurde.

1.17 RnCommTest V4.06

1.17.1 Hinweis auf die Hilfe

Beim erstmaligen Starten des Tools wird jetzt auf diese Hilfe hingewiesen.

1.17.2 Korrektur der Element-Größen

Bei manchen Anzeige-Einstellungen (insbesondere beim Standard-Schema unter Windows 10) wurden manche Fenster nicht in der korrekten Größe angezeigt und dadurch evtl. Controls „abgeschnitten“. Deshalb wurde bei allen Fenstern die Eigenschaft „AutoSize“ auf True gesetzt. Damit sollte dieses Problem nicht mehr auftreten.

Außerdem wurden bestimmter Controls verlängert (z.B. numerische Eingaben).

1.17.3 Ethernet – Korrektur wegen IpV6

Wird ein PC unter einem IpV6-Netzwerk (größerer Adressbereich) betrieben, gab es eine Exception beim Auslesen der Ip-Adresse.

Jetzt wird die Ip-Adresse zweiteilig ausgelesen („IpV4___IpV6“, z.B. „192.168.178.1___fe80:8ad:5a04:b0a9“).

1.17.4 OpcUa-Client – Korrektur

Bei manchen Servern konnte nicht korrekt gebrowst werden. Dies wurde behoben.

Hintergrund: Beim Browsen eines Knotens bekommt man dessen ReferenceDescription. Diese enthält auch die Adressierung, aber nicht als einfache NodeId, sondern als ExpandedNodeId, die zusätzlich die Server-Identifikation enthält. Zur korrekten Umwandlung in eine einfache NodeId wird bei manchen Knoten die Tabelle der verwendeten Namensräume („NamespaceTable“) benötigt. Da diese nicht berücksichtigt wurde, konnte bei manchen Servern nicht korrekt gebrowst werden. Jetzt werden alle Umwandlungen mit dieser Tabelle vorgenommen, wenn diese vorhanden ist.

1.17.5 OpcUaClient – Endpoint-New: Korrektur bei der Angabe der EndpointUrl

Wurde eine neue Endpoint-Url eingegeben und Enter gedrückt, wurde sie nicht gleich übernommen. Auch beim Verlassen mit Ok wurde sie nicht übernommen. Das ist jetzt behoben.

1.17.6 OpcUaClient – Endpoint-New: Angabe des Session-Timeouts

Bis jetzt wurde immer der Default-Session-Timeout verwendet. Jetzt kann explizit ein Session-Timeout angegeben werden.

1.17.7 OpcUaClient – Endpoint-Discover: Anzeige des Server-Zertifikats

Durch das Kontext-Menü kann jetzt das Zertifikat eines gefundenen Servers angezeigt werden.

1.17.8 OpcUaClient – Endpoint-Search: Korrektur wegen IpV6

Beim Aktualisieren der Adapter kam es zur selben Exception wie bei Ethernet (siehe oben).

1.17.9 OpcUa-Client – Neuer Reiter „ServerDiag“

Auf diesem Reiter ist eine komfortable Diagnose des Servers, den verbundenen Clients und deren Subscriptions möglich.

1.18 RnCommTest V4.05

1.18.1 Hilfe um den Anhang „ASCII-Codes“ erweitert.

In vielen Modulen ist das Kennen von ASCII-Codes hilfreich. Deshalb wurde diese Hilfe mit einer Tabelle erweitert.

1.18.2 Ethernet – Erweiterung: Anzeige des Adapter-Titels

Zusätzlich zu den bisher angezeigten Daten wird jetzt auch der Titel des Adapters angezeigt. Er erleichtert die Identifizierung.

1.18.3 Ethernet – Erweiterung: Suche im Netzwerk

Durch Pingen eines Netzwerk-Bereichs können Geräte identifiziert werden. Außerdem kann beim normalen Ping ein Timeout angegeben werden.

1.18.4 Pvi – Korrektur Browse-Fenster

Das Browse-Fenster wurde von den unteren Controls überdeckt.

1.18.5 OpcUaClient – Endpoint: Auswahl der Endpoint-URL

Für die Endpoint-URL gibt es jetzt ein Auswahl-Feld, in denen die letzten Eingaben gespeichert werden.

1.18.6 OpcUaClient – Endpoint: Angabe der bevorzugten Sprachen

Bei der Anmeldung des Clients kann jetzt auch eine Liste an bevorzugten Sprachen für sprachübersetzbare Texte (Datentyp „LocalizedText“) übergeben werden.

1.18.7 OpcUaClient – Endpoint: Angabe einer Session-Name-Erweiterung

Zur Unterscheidung mehrerer Instanzen des Tools an einem Server kann jetzt der Session-Name „RnCommTest“ durch eine Text-Angabe erweitert werden.

1.18.8 OpcUaClient – New Endpoint: Timeout für “Discover”

Es kann jetzt auch ein Timeout für den Discovery-Service angegeben werden.

1.18.9 OpcUaClient – New Endpoint: Mehr Discover-Angaben

Bei gefundenen Servern werden jetzt mehr Details angezeigt.

1.18.10 OpcUaClient – New Endpoint: Neuer Reiter “Search”

Auf diesem Reiter kann ein Netzwerk nach OpcUa-Servern durchsucht werden.

1.18.11 OpcUaClient – Zertifikats-Prüfung bei Connect

Server-Zertifikate, die in der Trusted- oder CA-Liste der eigenen Zertifikats-Verwaltung enthalten sind, wird jetzt automatisch vertraut. In der Ausgabe wird darauf hingewiesen.

1.18.12 OpcUaClient – DirectAccess.Advanced: Korrektur

Es werden jetzt auch Enum-Arrays korrekt dargestellt. Wenn möglich, werden die Integer-Werte in die Enum-Texte konvertiert.

1.18.13 OpcUaClient – DirectAccess.Advanced: Anzeige des CommonCode

Wird der Property-Knoten „EngineeringUnits“ eines Knotens angewählt, wird jetzt zusätzlich zur Anzeige der übertragenen Werte auch der aus der UnitId berechnete CommonCode (3-stelliger Text) angezeigt (siehe dazu auch den neuen Abschnitt unter Hinweise)

1.18.14 OpcUaClient – DirectAccess.Advanced: Anzeige eines Argument-Datentyps

Bei der Anzeige von In- oder Output-Argumenten einer Methode wird jetzt der Datentyp auch als Klartext angezeigt (z.B. „UInt32 (i=7)“ statt wie bisher „i=7“).

1.18.15 OpcUaClient – DirectAccess: Neuer Reiter “Translate”

Auf diesem Reiter kann durch Angabe einer Start-NodeId und eines relativen Browse-Pfades die absolute Ziel-NodeId übersetzt werden.

1.18.16 OpcUaClient – Attributes: Ermitteln der absoluten Browse-Pfade

Über den Kontext-Menü-Punkt „Get Absolute BrowsePathes“ können alle absolute Browse-Pfade des momentan selektierten Knotens ermittelt werden. Diese sind beim Parametrieren von anderen Clients manchmal sehr hilfreich.

1.18.17 OpcUa-Client – Subscription: Detailliertere Ausgabe bei Var-Items

Die Notification-Ausgaben eines Variablen-Items wurden überarbeitet. Es werden jetzt zusätzlich folgende optionale Ausgaben gemacht:

- LimitBits, wenn > 0
- QueueOverflow, wenn TRUE
- Status, wenn <> Good

1.18.18 OpcUa-Client – Subscription: Untergrenze Intervalle

Sowohl das Publishing-Intervall bei Subscription-Settings als auch das Sampling-Intervall bei Item-Settings kann jetzt auf -1 gestellt werden. Bei 0 sollte der Server auf das kleinstmögliche Intervall korrigieren. Bei -1 sollte der Server auf das Default-Intervall korrigieren.

1.18.19 OpcUa-Client – Subscription: Neue Einstellung für Items

Die neue Item-Einstellung „Filter Trigger“ gibt an, was auf Änderung überwacht wird: Status, Status/ Wert oder Status/Wert/Zeitstempel.

1.18.20 OpcUa-Client – Subscription: Triggerung von Items

Ein Item kann jetzt als Trigger für die Übertragung eines oder mehrerer anderer Items eingestellt werden.

1.18.21 OpcUa-Client – Certificates: Größere Schlüssel-Länge möglich

Bei Neuerstellung eines Zertifikats kann jetzt die Schlüssel-Länge in folgenden Stufen angegeben werden: 1024, 2048, 4096, 8192, 16384, 32768, 65536.

1.18.22 OpcUa-Client – Certificates: Angabe des Signatur-Algorithmus

Bei Neuerstellung eines Zertifikats kann jetzt der Signatur-Algorithmus (sha1RSA oder sha256RSA) angegeben werden.

1.18.23 OpcUa-Client – Hinweise: Umrechnung von UnitId zu CommonCode

Die Umrechnung von UnitId zu CommonCode und zurück wurde als Hinweis aufgenommen.

1.19 RnCommTest V4.04

1.19.1 OpcUaClient – Discover

Es kann jetzt eine maximale Anzahl der gefundenen Endpoints angegeben werden, um die Suche zu verkürzen.

1.19.2 OpcUaClient – Erweiterte Einstellungen

In den Endpoint-Eigenschaften kann jetzt ein Dialogfeld mit erweiterten Einstellungen geöffnet werden, welche Parameter für die Diagnose enthalten.

1.19.3 OpcUaClient – Connection: Neue Einstellung

Durch die neue Einstellung „Return Diagnostics Mask“ kann festgelegt werden, welche Diagnose-Informationen vom Server bei jeder Anfrage angefordert werden.

1.19.4 OpcUaClient – DirectAccess.Advanced: Korrektur

Die Liste wird jetzt gelöscht, wenn der Wert des selektierten Nodes nicht ausgelesen werden kann.

1.19.5 OpcUaClient – DirectAccess.Advanced: Speichern und Laden von Werten

Durch das Kontextmenü können jetzt Werte (auch komplexe Strukturen) in einer Datei gespeichert und später wieder geladen werden.

1.19.6 OpcUaClient: Abfangen von NULL-Werten

Durch eine Änderung am Stack werden Nodes jetzt auch korrekt behandelt, wenn Standard-Attribute vom Server als NULL angegeben werden (z.B. ‚UserWriteMask‘). Vorher wurde eine Exception ausgelöst und dadurch die Knoten nicht angezeigt.

1.19.7 OpcUa-Client – DirectAccess.Advanced: Anzeige von irregulären Enumerationsen

Bei irregulären Enumerationsen (nicht bei 0 beginnend oder nicht durchlaufend nummeriert) können jetzt die definierten Werte und die dazugehörigen Texte über den Datentypen ermittelt und so auch hier eine Auswahl angeboten werden.

1.19.8 OpcUa-Client – Subscription: Korrektur

Wenn die Option ‚Allow restricted operations‘ gesetzt ist, kann jetzt auch in der Subscription ein Wert eingegeben werden, der dann anschließend geschrieben wird.

1.19.9 OpcUa-Client – MethodCall: Anzeige des Pfades

In der Statuszeile wird jetzt der Pfad zur selektierten Methode angezeigt.

1.19.10 OpcUa-Client – MethodCall: Enumeration als Argument

Ist ein Argument eine Enumeration, so wird jetzt auch dieser Datentyp aufgelöst.

1.19.11 OpcUaClient – MethodCall: Unterstützung von komplexen Argumenten

Es werden jetzt auch Extension-Objects-Argumente (z.B. Strukturen) unterstützt.

1.19.12 OpcUaClient – MethodCall: Speichern und Laden von Einzel-Argumenten

Analog zur Anzeige von Werten können jetzt auch komplexe Eingangs-Argumente in einer Datei gespeichert und später wieder geladen werden.

1.19.13 OpcUaClient – MethodCall: Speichern und Laden von Argument-Listen

Die Eingangs-Argumente können jetzt komplett in einer Datei gespeichert und später wieder geladen werden.

1.19.14 OpcUaClient – MethodCall: Behandlung von generischen Arrays

Sind in einem Struktur-Eingangs-Argument Arrays vorhanden, so kann jetzt die Anzahl der Elemente verändert werden.

1.19.15 OpcUa-Client – Stack-Korrektur

Bei RequestHeadern wurde der TimeoutHint manchmal mit 0 oder sehr kleinen Werten besetzt. Dies löste bei manchen Servern einen Timeout aus.

1.20 RnCommTest V4.03

1.20.1 OpcUaClient – Attributes: Anzeige von lokalisierten Texten

Ein Wert vom Typ „LocalizedText“ beinhaltet zwei Angaben: das Kürzel der Sprache (z.B. „en“ für Englisch oder „de“ für Deutsch) und den eigentlichen Text. Wenn ein Kürzel angegeben ist, wird jetzt die Ausgabe folgendermaßen formatiert:

Kürzel:Text

Ist das Kürzel „null“ oder ein Leerstring, wird wie bisher nur der Text ausgegeben.

1.20.2 OpcUa-Client – DirectAccess.Advanced: Anzeige eines Fortschrittsbalkens

Das Umsetzen eines Wertes (z.B. langer Byte-Arrays) in die Liste braucht u.U. einige Zeit. Um dem Benutzer anzuzeigen, dass das Tool daran arbeitet, wird jetzt eine Meldung mit Fortschrittsbalken angezeigt.

1.20.3 OpcUa-Client – DirectAccess.Advanced: Anzeige von Byte() und ByteString

In der Spalte „Description“ wurde auch immer die ASCII-Interpretation angezeigt. Dies wird jetzt nicht mehr gemacht, weil es signifikante Performance-Verbesserung bringt.

Um die Text-Interpretation trotzdem zu sehen, kann der Kontext-Menüpunkt „Show as text“ benutzt werden.

1.20.4 OpcUa-Client – DirectAccess.Advanced: Dialog zum Anzeigen von Texten

Angezeigte Texte können bei ReadWrite-Items auch geändert werden. Das Zurückwandeln sehr langer Texte in den ursprünglichen Datentyp (z.B. Byte-Array) kann sehr lange dauern.

Deshalb hat das Dialogfeld jetzt auch einen „Cancel“-Button, bei dem die Rückwandlung entfällt.

1.20.5 OpcUa-Client – Subscription: Korrektur

Bei einem ReadOnly-Struktur-Item wurde bei einem Doppelklick nicht das Wert-Dialogfeld eingeblendet.

1.20.6 OpcUa-Client – Subscription: Korrektur

Wenn zusätzlich zu einem Struktur-Item noch weitere Items vorhanden waren und das Wert-Dialogfeld dieser Struktur geöffnet war, kam es zu einer Ausnahme und die Werte wurden nicht korrekt angezeigt.

1.20.7 OpcUa-Client – MethodCall: Korrektur

Manchmal wurde das Parent-Objekt nicht richtig ermittelt, das zur eindeutigen Adressierung einer Methode benötigt wird. Das führte dann zu der Statusmeldung „0x80330000 = Bad_NodeIdInvalid“.

1.21 RnCommTest V4.02

1.21.1 OpcUa-Client – Behandlung von ExtensionObjects (z.B. Strukturen)

Es werden jetzt komplexe Datentypen (ExtensionObjects, z.B. Strukturen) unterstützt, welche als Byte-Array gelesen und geschrieben werden (Binary Encoding). Sie werden mit vom Server gelieferten Informationen in ihre ursprüngliche Form konvertiert.

1.21.2 OpcUaClient – Browse: Änderung des Browsings

Die Checkbox „Show Object Types“ wurde ersetzt durch die Checkbox „Show NonHierarchical References“. Dadurch ändern sich die dargestellten Knoten in beiden Modis.

1.21.3 OpcUaClient – Attributes: WriteMask als Text

Die Werte der numerischen Attribute „WriteMask“ und „UserWriteMask“ werden jetzt zusätzlich als textuelle Auflistung dargestellt.

1.21.4 OpcUaClient – Attributes: Neuer Menüpunkt „Go to Node“

Wenn der Wert eines Attributes einen Knoten darstellt (z.B. beim Attribut „DataType“), kann mit dem Kontext-Menüpunkt „Go to Node“ der Knoten gesucht und automatisch selektiert werden.

1.21.5 OpcUa-Client – Access: Dialog für Texte

Für Strings und Byte-Arrays kann jetzt ein Textfenster angezeigt werden. Das ist besonders für lange Strings komfortabel und nützlich bei sogenannten Data-Dictionaries, welche Datentyp-Informationen enthalten.

1.21.6 OpcUa-Client – DirectAccess.Advanced: Kopieren in die Zwischenablage

Aus diesen Listen können jetzt Wert-Texte per Kontext-Menü oder die Tastenkombination „Ctrl-C“ in die Zwischenablage kopiert werden.

1.21.7 OpcUa-Client – Neuer Reiter „References“

Auf diesem Reiter werden die Referenzen eines Knotens dargestellt.

1.21.8 OpcUa-Client – Subscription: Korrigieren der Settings durch den Server

Durch eine entwicklungsseitige Änderung des Stacks wurden vom Server korrigierte Werte (z.B. das SamplingInterval des Items) nicht mehr auf das Item zurück übernommen. Dies wird jetzt vom Tool gemacht. Dadurch kann jetzt auch eine Meldung ausgegeben werden, welche Werte korrigiert wurden.

1.21.9 OpcUa-Client – MethodCall: Anzeige und Editieren von generischen Arrays

Manche System-Methoden haben generische Arrays als Eingangs-Argumente. Das bedeutet, dass die Länge des Arrays nicht statisch ist, sondern vom Aufrufer festgelegt wird. Alle generischen Arrays von Standard-Datentypen werden jetzt unterstützt.

1.21.10 OpcUa-Client – MethodCall: Anzeige von komplexen Output-Argumenten als Byte-Array

Hat eine Methode einen unbekannten Datentyp als Output-Argument, so wird dieses als Byte-Array übertragen. Solche Argumente werden jetzt auch als Byte-Array (mit Ascii-Übersetzung) angezeigt.

1.21.11 OpcUa-Client – Neuer Reiter „Performance“

Auf diesem Reiter können Zeitmessungen für Lese- und Schreib-Operationen gemacht werden.

1.21.12 OpcUa-Client – Certificates: Anzeige von erweiterten Feldern

Bei den Zertifikaten werden jetzt nicht nur die Standard-Felder, sondern auch die erweiterten Felder angezeigt.

1.22 RnCommTest V4.01

1.22.1 OpcUa-Client – Kopieren in die Zwischenablage

Aus diversen Listen können jetzt Wert-Texte per Kontext-Menü oder die Tastenkombination „Ctrl-C“ in die Zwischenablage kopiert werden.

1.22.2 OpcUa-Client – Erweiterung des Attribut-Zugriffs

Der Zugriff ist jetzt auf drei Reiter-Seiten verteilt: „Standard“, „Advanced“ und „Direct“. Der erste enthält den Zugriff wie bisher, der zweite erlaubt den Zugriff auf erweiterte Objekte, z.B. Arrays oder Enumerationen, der dritte erlaubt die Adressierung ohne Browsing.

1.22.3 OpcUa-Client – Subscription

Zur manuellen Änderung des Werts eines abonnierten Knotens wird jetzt ein Dialog verwendet, wenn der Wert kein Standard-Datentyp ist, z.B. bei Arrays.

1.22.4 OpcUa-Client - Neue Endpoint-Einstellung

Es gibt jetzt eine Einstellung, damit der gelieferte Hostname durch die IP-Adresse ersetzt wird. Das erleichtert die Adressierung, wenn kein DNS-Server verfügbar ist.

1.22.5 OpcUa-Client - Neue Client-Einstellung

Es kann jetzt angegeben werden, ob das Tool Enumerationen als Int32 anzeigt wie bisher oder ob es versuchen soll, die Werte als Text zu interpretieren.

1.22.6 OpcUa-Client - Neue Find-Einstellung

Beim Suchen innerhalb des Browse-Baums kann jetzt der Knoten ‚DeviceSet‘ übersprungen werden.

1.22.7 OpcUa-Client – Stack-Änderung

Die Akzeptanz eines Server-Zertifikats wurde bisher nur bei verschlüsselter Verbindung abgefragt (siehe Server-Einstellung „Accept Server Certificate Automatically“). Jetzt wird dies auch bei einer unverschlüsselten Verbindung gemacht.

1.22.8 OpcUa-Client – Stack-Änderung

Wird ein Zertifikat auf nicht gültig validiert, wird jetzt eine genauere Ursache angegeben.

1.22.9 OpcUa-Client - Verbindung

Bei einem nicht akzeptierten Server-Zertifikat wird jetzt ein Dialogfeld mit genauerer Ursache eingeblendet.

1.22.10 OpcUaClient – Verbindung

Für die Discovery-IP gibt es jetzt ein Auswahl-Feld, in denen die letzten Eingaben gespeichert werden.

1.23 RnCommTest V4.00

1.23.1 Umstellung der Entwicklungsumgebung

Es wurde von MS Visual Studio 2008 auf MS Visual Studio 2015 umgestellt. Das war nötig, um den neuesten OpcUa-Stack implementieren zu können.

1.23.2 Umstellung des Frameworks

Es wurde von DotNet-Framework V3.5 auf V4.5.1 umgestellt. Das war nötig, um den neuesten OpcUa-Stack implementieren zu können.

1.23.3 Umstellung des OpcUa-Stacks

Es wurde von OpcUa-Stack V1.02.334 auf V1.03.340 umgestellt.

1.23.4 OpcUa-Client - Neue Endpoint-Einstellung

Es gibt jetzt eine Einstellung, um das clientseitige Prüfen des Host-Namens zu verhindern („Check Server Hostname“).

Bei einigen B&R-SPS-Betriebssystemen wird vom Server ein falscher Hostname („br-automation“ statt IP-Adresse) geschickt. Das führte beim neuen Stack zu einer Exception. Durch die neue Einstellung kann diese Überprüfung ausgeschaltet werden.

1.23.5 OpcUa-Client – Änderung und Erweiterung für Events

Die Behandlung der Event-Items in einer Subscription wurde überarbeitet. Insbesondere das Setzen eines Filters (SelectClause) wurde verbessert. So können nun alle am Server verfügbaren Event-Typen (z.B. Audit-Event-Typen) einzeln ausgewählt werden.

1.23.6 OpcUa-Client – Änderung und Erweiterung für HistoricalAccess

Die Behandlung des HistoricalAccess wurde für den B&R-Server überarbeitet. Außerdem gibt es jetzt eine zyklische Lese-Funktion.

1.23.7 OpcUa-Client - Monitor

Aufgrund der Erweiterungen der letzten Versionen (vermutlich seit V3.10) funktionierte der Monitor nicht mehr richtig. Diese Fehler wurden behoben.

1.24 RnCommTest V3.12

1.24.1 OpcUaClient – MethodCall

Es wurden Korrekturen sowie neue Funktionalitäten implementiert:

Der Timeout wird jetzt an den Server übergeben. Es gibt Server, die diesen Wert nicht verwenden. Wenn doch, wird vom Server nach dieser Zeitspanne der Status „BadRequestTimeout“ zurückgeliefert. Wenn also eine 0 übergeben wird wie bisher, liefert ein solcher Server immer einen Fehler.

Der Timeout kann jetzt für jede Methode separat geändert werden.

Als Argumente werden jetzt auch multidimensionale Arrays unterstützt.

1.24.2 OpcUaClient – ServerInfo

Eine zusätzliche Reiter-Seite zeigt nun Informationen zum verbundenen Server an.

1.24.3 OpcUaClient – Zertifikate

Es wurde eine Zertifikats-Verwaltung implementiert. Es ist nun möglich, selbst Zertifikate zu erstellen und eines davon zu verwenden.

1.25 RnCommTest V3.11

1.25.1 OpcUaClient – Verbindung

Für den Benutzernamen und -passwort gibt es jetzt Auswahl-Felder, in denen die letzten Eingaben gespeichert werden.

1.25.2 OpcUaClient – Korrektur

Beim Schreiben eines Werts außerhalb der „EU-Range“ wird jetzt eine Fehlermeldung ausgegeben, wenn der Wert nicht angenommen wurde.

1.25.3 OpcUaClient – MethodCall

Der Dialog aus dem Browse-Kontext-Menü wurde entfernt. Dafür gibt es jetzt einen eigenen Reiter mit mehr Features.

1.26 RnCommTest V3.10

1.26.1 OpcUaClient – Verbindung

Bis jetzt war es so, dass vor einem Verbindungsaufbau alle Anmelde-Daten mit den Daten aus dem Discovery-Dienst abgeglichen wurden, auch die Endpoint-Url.

Wenn also eine Endpoint-Url manuell eingegeben wurde, der Discovery-Dienst diese aber falsch lieferte, konnte keine Verbindung aufgebaut werden.

Jetzt wird immer die eingegebene Endpoint-Url verwendet.

1.26.2 OpcUaClient – Discovery

Die Discovery-Daten (Hostname und Port) werden jetzt auch in der *.ini-Datei gespeichert und bleiben somit nach einem Neustart erhalten.

1.26.3 ModbusMaster – Limitierung der Wert-Eingabe

Für einen MbValue war die Wert-Eingabe auf -32768..+32767 limitiert. Leider konnten damit keine Werte eingegeben werden, welche zur Interpretation als Float (ein Float oder ein Real besteht aus 4 Bytes) gebraucht werden. Deshalb ist die Eingabe jetzt auf -32768..+65535 limitiert.

1.26.4 Pvi – Default-Aktivierung

In V3.09 war das Pvi-Modul standardmäßig aktiviert. Das gibt Fehlermeldungen auf Rechnern, bei denen Pvi nicht installiert ist. Jetzt ist es standardmäßig wieder deaktiviert.

1.27 RnCommTest V3.09

1.27.1 Allgemein – Fenster-Größe

Das Haupt-Fenster des Tools kann jetzt in der Größe verändert werden. Die eingestellte Größe bleibt nach einem Neustart erhalten. Die Größe des Inhalts wird dabei skaliert. Die meisten Inhalts-Bereiche können ebenfalls in ihrer Größe verändert werden, um eine übersichtlichere Darstellung zu ermöglichen. Diese Größen-Werte werden aber nicht gespeichert.

1.27.2 OpcUaClient – Attribute-Anzeige

Es werden jetzt in der Attribute-Liste ebenfalls sowohl der Source- als auch Server-Zeitstempel angezeigt.

1.27.3 OpcDaClient – Reiter wird versteckt

Das Modul „OpcDaClient“ (klassischer Opc) wird jetzt standardmäßig versteckt. Es kann in der Ini-Datei durch den Eintrag „WithOpcDaClient“ wieder aktiviert werden. Dieses Modul wird nicht mehr weiterentwickelt und gewartet.

1.28 RnCommTest V3.08

1.28.1 OpcUaClient – Discovery

Beim Discovery kann jetzt auch eine Portnummer angegeben werden.

1.28.2 OpcUaClient – Attribute-Anzeige

Es werden jetzt in der Detail-Anzeige sowohl der Source- als auch Server-Zeitstempel angezeigt.

1.28.3 OpcUaClient – Subscription

Die Vermischung von Strukturen und Arrays wird jetzt beim Aufsplitten richtig behandelt.

1.28.4 OpcUaClient – Subscription - ItemSettings

Für Variablen kann jetzt eingestellt werden, ob der Source- oder der Server-Zeitstempel angezeigt bzw. im Trace verwendet werden soll.

Für ein Event kann jetzt ein komplexer Filter gesetzt werden.