

Bristol OpenEnterprise Reference Guide

NW3000 Administrator Guide

Contents

1 Network 3000 Administrator Guide..... 3

1.1 Database Build..... 3

1.1.1 Configuring Post DBB and TPB Scripts 3

1.1.2 Alarms..... 4

1.1.2.1 ACCOL Version Mismatch Alarm Conditions 4

1.1.2.2 Using the #PWRUP Remote Alarm 5

1.1.2.2.1 #PWRUP and Template Collection 5

1.1.2.2.2 #PWRUP and RBE Initialization 5

1.1.2.3 Purging the Alarm Summary Table..... 6

1.2 Data Collection..... 6

1.2.1 Preventing Stress on the BSAP Network 6

1.2.2 Collection By Templates..... 7

1.2.2.1 Creating Timeclasses..... 8

1.2.2.1.1 Standard Timeclasses - Worked Example 8

1.2.2.1.2 One-Shot Timeclasses 8

1.2.2.2 Modifying Timeclass Collection Frequencies..... 8

1.2.2.3 Setting up Template Collection 8

1.2.2.4 Assigning Timeclasses 9

1.2.2.5 Logical Alarm Template Collection 9

1.2.2.6 Manually Triggering Template Collection 10

1.2.2.6.1 Manual Collection of All Templates 10

1.2.2.6.2 Manual Collection of Specific Templates..... 10

1.2.3 Collection By RBE 10

1.3 NW3000 Driver (RDI3000)..... 10

1.3.1 Auto Starting OpenBSI Using RDI3000..... 10

1.3.1.1 Set the NW3000Driver 11

1.3.1.2 Set the RDI3000 Registry Key 11

1.3.2 Detecting Devices Going Offline..... 11

1.3.3 RDI3000 Forcelock Checking and Timeout..... 12

1.3.3.1 Configuration 12

1.4 Time Zone Issues..... 12

1.4.1 Time Zones and Daylight Saving Time 12

1.4.2 Disabling Timestamp Adjustment..... 13

1.4.3 OpenBSI Time Synchronization 13

2 Index 14

1 Network 3000 Administrator Guide

This documentation is primarily for System Administrators. It explains the configuration options that can be used to modify and fine tune the default behaviour of OpenEnterprise with regard to Bristol Babcock RTUs and their native driver software suite, OpenBSI.

Set up details for the OpenEnterprise NW3000 Database Builder, NW3000 Poll List Builder and the NW3000 Driver (RDI300) are covered under the following headings:-

1. Database Build
2. Data Collection
3. NW3000 Driver
4. Time Zone Issues

Although targeted at System Administrators of OpenEnterprise SCADA systems, this documentation does not attempt to fully explain the functionality and use of Network3000 or ControlWave RTUs within OpenEnterprise.

1.1 Database Build

OpenEnterprise builds NW3000 and ControlWave signals and poll lists into the OpenEnterprise database automatically using the NW3000 Database Builder (DBB) and Poll List (or Template) Builder (TPB). This section deals with the set-up of the DBB and TPB generally and with regard to remote alarms.

1.1.1 Configuring Post DBB and TPB Scripts

Both DataBase Builder (DBB) and Poll List Builder (TPB) can be optionally configured to run customised scripts following a database or template build.

The intention is that these scripts are custom SQL statements to be run following a build. However, anything can be run, not just SQL scripts.

The scripts to run are configured in the INI files DBB.INI and TPB.INI respectively. These files are optional and should reside in the default Windows™ directory.

The following is an example DBB.INI file. The [postbuild] section specifies the default script details. The [postbuild:<dataservice>] specifies the optional dataservice specific script details that override those specified in the [postbuild] section. The 'visible' value makes the script run with desktop interaction and should be considered as just a debug option when checking if the script has been configured properly. Only the 'target' value is required.

In this example where DataBase Builder is connected to database "rtrdb1", DataBase Builder runs the SQL Client which automatically loads the SQL file POSTDBB2.SQL, with output being captured in the file DBB_SQL.TXT.

```
[postbuild]
target=C:\Program Files\Bristol\OpenEnterprise\Bin
args=-r include_file=postdbb.sql sql > sql.txt
```

startin=C:\Program Files\Bristol\OpenEnterprise\Data

visible=1

[postbuild:rtrdb1]

args=-r include_file=postdbb2.sql sql > dbb_sql.txt

The TPB.INI file has an identical format to the DBB.INI file.

1.1.2 Alarms

Device Failure alarm conditions enable the OpenEnterprise Database to flag an alarm should there be any communication failure with the device.

Device status alarm conditions are automatically created when the DBB process creates a new device. Entries are inserted into the nw3000DeviceStatusAlarmCondition table. The following attributes are set as follows:

- Priority = 252 (where Priority is the alarm condition's priority).
- Condition = 9 (where 9 is a 'Change of State' condition type).
- Devicename = nw3000device.devicename (which is the RTU's devicename).

Device failure alarms use the alarm summary as follows, where <> denotes the value will be substituted and | denotes a set of possible values.

Attribute	Values
Name	<devicename>
Description	COMMUNICATIONS FAILURE <devicename>
Devicename	<devicename>
Alarmtext	<devicename> FAILED OKAY
Value	FAILED OKAY
Eventtype	COMMUNICATIONS STATUS

The automatic creation of device status alarm conditions will be inhibited if the poly.cfg resource oe_nw3000_supress_status_alarms is set to TRUE. The default value of the resource is FALSE.

1.1.2.1 ACCOL Version Mismatch Alarm Conditions

When the DataBase Builder and TempPlate Builder processes are invoked they use the information from the ACCOL load file found in the ACCOL directory on the OpenEnterprise Server.

If subsequently a new ACCOL file is downloaded to the RTU(s), then the OE Database will detect an ACCOL version mismatch.

Device status alarms include the annunciation of an Accol version mismatch for Network 3000 devices. When a version mismatch exists, template collection may fail and RBE collection will be less efficient.



A correctly configured server should have both DBB and TPB running in monitor mode. If a version mismatch occurs due to a download from the server, then DBB and TPB will automatically resolve the mismatch.

Even so, version mismatches may still persist if ACCOL files have been downloaded locally at the RTU. When this occurs the DBB will resolve the database MSD store from the RTU and then the TPB will automatically rebuild the database template store.

However, because the OE Server does not have access to the new ACCOL load file, changes made to the ACCOL signal configuration will not be changed in the OE Database. This will cause a persistent version mismatch.

Therefore, when a device is inserted into the database (i.e. through the DBB process) optional ACCOL version mismatch alarm conditions will be generated in the `nw3000DeviceStatusAlarmCondition` table.

The automatic creation of ACCOL version mismatch alarm conditions will be inhibited if the `poly.cfg` resource `oe_nw3000_supress_version_alarms` is set to TRUE. The default value of the resource is FALSE.

1.1.2.2 Using the #PWRUP Remote Alarm

The system can be configured to monitor for #PWRUP.000. remote alarms being generated by NW3000 devices. The receipt of a #PWRUP alarm can optionally be used to generate one or more of the following actions.

- Request to collect all templates for the NW3000 device.
- Request initial values for all RBE signals for the NW3000 device.

NOTE: This functionality only applies to a NW3000 device that generates a #PWRUP alarm while OpenEnterprise believes that the NW3000 device is alive. Receiving a #PWRUP alarm from a currently 'dead' NW3000 device does not result in any special #PWRUP logic being executed.

1.1.2.2.1 #PWRUP and Template Collection

The receipt of the #PWRUP alarm results in RDI 3000 requesting a template collection for all templates configured for the appropriate RTU. This option is strongly advisable if the user is using one-shot templates or slow collecting templates.

The ***EnablePWRUPtpls*** value is found on the ***OpenEnterprise\Tasks\RDI3000*** key, and the option is enabled by default. To disable the functionality, use the Settings Editor to set the value data to 0 (zero).

1.1.2.2.2 #PWRUP and RBE Initialization

The receipt of the #PWRUP results in the NW3000 RDI sending an initial values request for all RBE values to the appropriate NW3000 device. This should not be necessary, as in most scenarios the NW3000 device should generate a 'going active' message.

The ***EnablePWRUPRbe*** value is found on the ***OpenEnterprise\Tasks\RDI3000*** key, and the option is enabled by default. To disable the functionality, use the Settings Editor to set the value data to 0 (zero).

1.1.2.3 Purging the Alarm Summary Table

By default, the alarm summary will be automatically purged on a per device basis when the remote ACCOL load version changes.

To disable this default behaviour set the `nw3000Device.DisableAutoPurge` to TRUE.

A manual purge of all alarms for a given device can be performed by setting the `nw3000Device.AlarmPurge` attribute to TRUE.

1.2 Data Collection

The database supports (by design) the update of primary key values. The modification of NW3000 primary key values can cause problems for data collection and the viewing of data at the workstation.

The nw3000 CL has been enhanced to prevent the update of primary key values for the following tables:

- Nw3000Driver_table.
- Nw3000Network_table.
- Nw3000Device_table.
- Nw3000RealAnalog_table.
- Nw3000Digital_table.
- Nw3000PollList_table.

Any transaction that attempts to modify the primary key of any of the above tables will result in an error and the transaction will be aborted.

This functionality is controlled by a new Poly.cfg resource, `oe_nw3000_suppress_pk_update`, which will have a default value of TRUE. Setting this resource to FALSE will allow primary key modifications.

Note that this implementation is Network 3000 specific. The update of Allen Bradley primary key values will be unaffected by this. In NW3000, the signal name (primary key) is a unique identifier for a signal within an ACCOL load and hence has meaning within the device. For Allen Bradley PLCs, the signal name identifier has no meaning within the PLC and hence updates may be desirable.

1.2.1 Preventing Stress on the BSAP Network

When OpenEnterprise starts, or when a fail-over occurs, it is possible for the BSAP network to become oversubscribed with request messages from the NW3000 RDI. These messages will primarily be initial template requests but will also include RBE initialisation requests and any user updates. This can be particularly evident on slow communication networks with large numbers of devices.

Request message throttling can be enabled within OpenEnterprise in order to limit the maximum number of outstanding requests that can be submitted by the NW3000 RDI.

One can set the Nw3000Driver.TotalRequests attribute to the maximum number of outstanding requests the user may want at any one time. When set to 25 (for example) the NW3000 RDI will start throttling requests when there are 25 outstanding. These throttled requests are added to a list of pending requests and are sent when outstanding requests are completed. No request messages will be discarded or lost.

One can monitor how many request messages the NW3000 RDI has throttled by querying the nw3000Driver.numthrottled value when network statistics are enabled. Note that numthrottled is a total not a snap shot. Note also that enabling network statistics also causes nw3000device.pollsdone, writesdone, pollsfailed and writesfailed to be updated for all devices.

e.g.

```
SQL> update nw3000driver set totalrequests=50;
```

Transaction Done

```
SQL> update nw3000network set statsfrequency=seconds(10);
```

Transaction Done

```
SQL> select numthrottled from nw3000driver;
```

```

+-----+-----+
| totalrequests | numthrottled |
+-----+-----+
| 50          |          9 |
+-----+-----+

```

Query Done: 1 record selected

In the above example, the SQL update statements set total requests to 50 and enables network statistics. The statistics will be updated by the NW3000 RDI once every 10 seconds. The SQL select statement shows that at the time of issuing the SQL select there had been 9 request messages that had been throttled.

If TotalRequests is set to NULL, the default, then it will be set to three quarters of the total wait packets configured for OpenBSI.

1.2.2 Collection By Templates

Template collection is when data is collected from the RTU at regular time periods (known as [Timeclasses](#)). Templates are also known as Poll Lists. When an RTU is marked for collection by Templates, the RDI3000 manages the collection of data by using lists of signals. These lists are called Templates and are created using the Poll List Builder. When Template collection is triggered, the data is requested from the RTU one timeclass at a time. This greatly improves the speed and efficiency of the data collection.



1.2.2.1 Creating Timeclasses

A timeclass is created by inserting entries into the `dvi_schedule` table. This can be done by using the relevant command line options with the NW3000 Database Builder, or by using the SQL Client. The user must specify the timeclass number (1 to 32767), the collection interval and an optional offset. If the user has several timeclasses configured then using an offset value is recommended as this will optimise the data collection process.

Timeclasses can be either standard timeclasses or one-shot timeclasses. A standard timeclass results in templates being collected at the rate specified. One-shot timeclasses have no rate and will only collect at start-up, when a previously dead device comes online, when templates are built or at operator request.

1.2.2.1.1 Standard Timeclasses - Worked Example

The following SQL statements will create timeclass 1 and 2, both collecting every 10 seconds. Timeclass 2 is offset by 5 seconds so should never collect at the same time as timeclass 1; i.e. timeclass 1 and timeclass 2 should be collected 5 seconds apart.

```
INSERT INTO dvi_schedule (scheduleid, maxinterval) VALUES
(1, seconds(10));

INSERT INTO dvi_schedule (scheduleid, maxinterval, offset) VALUES
(2, seconds(10), seconds(5));commit;
```

Note that timeclass entries must be created prior to performing a template build.

1.2.2.1.2 One-Shot Timeclasses

In order to create a one-shot timeclass the user needs to first create timeclasses as above but specify the `maxinterval` to be `seconds(0)`.

```
INSERT INTO dvi_schedule (scheduleid, maxinterval) VALUES
(3, seconds(0));commit;
```

1.2.2.2 Modifying Timeclass Collection Frequencies

The user can modify timeclass collection frequencies at runtime by updating the appropriate `dvi_schedule.maxinterval` field.

For example, the following SQL statement will modify the timeclass 1 collection frequency from 10 seconds to 5 seconds.

```
UPDATE dvi_schedule SET maxinterval = seconds(5) WHERE scheduleid = 1;
```

If RDI 3000 is running then this change will be reflected immediately. There is no need to stop and re-start RDI 3000.

1.2.2.3 Setting up Template Collection

In order to setup template data collection the user will need to perform the following steps.

- Build the Network 3000 devices and signals into the database using DataBase Builder (DBB).

- Assign timeclasses to devices and/or signals. The user can either configure signals into templates by setting individual signals timeclass values or by setting the default timeclass for the device.
- Create the time classes.
- Build templates.

1.2.2.4 Assigning Timeclasses

Default timeclasses can be assigned to devices. This means that all non-RBE signals within that device will use the default timeclass. To set the default timeclass the user will need to set the `nw3000Device.timeclass` attribute. This can either be done through DBB, the NW3000 Configuration Tool or SQL utilities.

e.g.

```
UPDATE nw3000device SET timeclass = 1 where devicename = 'CFE1';
```

However, if the user requires better control as to which individual signals are in which timeclasses then the user can override the devices default timeclass value by assigning timeclass values to individual signals. To set a signal's individual timeclass the user will need to set the `nw3000RealAnalog.timeclass` or `nw3000Digital.timeclass` attributes.

e.g.

```
UPDATE nw3000RealAnalog SET timeclass = 1 where name =  
CFE1:ANALOG.INPUT.001';
```

If the user wishes to use the default timeclass and have a set of one or more signals which are not collected by templates, then assign those signals to timeclass 0 (zero).

The order of precedence when assigning signals into timeclass templates is as follows:

1. The signal's timeclass value. If set to zero then the signal will never appear in a template. If set to a value between 1 and 32767 inclusive then the signal will always be collected by this timeclass. This overrides the device's default timeclass. If set to NULL then the device's default timeclass will be used.
2. The device's default timeclass.

1.2.2.5 Logical Alarm Template Collection

If the user wishes to collect logical alarms using a slow timeclass frequency then: -

- Create timeclass 254.
- Build templates.

By default, all digital remote alarms will be built into timeclass 254. If the user does not wish to collect these signals using timeclass 254 then set the signal's timeclass value to 0 (zero) or another valid timeclass value.

1.2.2.6 Manually Triggering Template Collection

Manual collection of templates is possible by rebuilding the templates using TPB. However, there are two other ways in which this may be done without a rebuild.

1.2.2.6.1 Manual Collection of All Templates

The CollectTemplates attribute, of type boolean, exists in the nw3000device table. The default value for CollectTemplates is FALSE.

This attribute will be monitored by RDI3000. When set to true, RDI3000 will initiate a template request for all templates owned by the relevant device.

1.2.2.6.2 Manual Collection of Specific Templates

The nw3000pollist.forcepoll attribute is supported for manual template collections. This will allow one or more templates to be manually collected.

e.g.

```
UPDATE nw3000pollist set forcepoll = true where devicename = <devicename>
and scheduleid = <timeclass>;
```

1.2.3 Collection By RBE

By default, RBE signals will not be built into templates.

If, in addition to collecting signal(s) via RBE, the user wants to collect the same signals via templates then set the signal's timeclass value to the appropriate timeclass value.

For example, the following two SQL statements will result in RBE signals for device 'CFE1' being built into timeclass 1.

```
UPDATE NW3000RealAnalog_Table SET Timeclass = 1 WHERE Rbe = TRUE and
devicename = 'CFE1'; commit;
```

```
UPDATE NW3000Digital_Table SET Timeclass = 1 WHERE Rbe = TRUE and
devicename = 'CFE1'; commit;
```

1.3 NW3000 Driver (RDI3000)

The RDI3000 can be configured to operate in different ways with regard to OpenBSI and Bristol Babcock devices:-

1. Auto Starting OpenBSI Using RDI3000
2. Detecting Devices Going Offline
3. RDI3000 Forcelock Checking and Timeout

1.3.1 Auto Starting OpenBSI Using RDI3000

The user can optionally configure the system such that Open BSI is started and stopped automatically by RDI3000.

1.3.1.1 Set the NW3000Driver

The following SQL statement will configure RDI 3000 to automatically start Open BSI when the system is active (standalone or master). It will start the Open BSI network definition file C:\Accol\current.ndf. When the RDI closes down it will also shut down Open BSI.

```
UPDATE nw3000driver SET obsistartup = 1, obsistartndf = 'C:\Accol\current.ndf', obsishutdown = true;
```

The above update is made when the [Automatically start OpenBSI](#) box is checked on the *OpenBSI AutoStart* page of the *System Setup Wizard* of the NW3000 Setup tool from the Toolbox, and also when the [Auto Start box](#) is checked on the *Driver Property page* of the NW3000 Advanced Setup tool.

Where obsistartup indicates whether RDI 3000 will start Open BSI, obsistartndf indicates the NDF file that Open BSI should open and obsishutdown tells RDI3000 whether to close Open BSI on shutdown.

1.3.1.2 Set the RDI3000 Registry Key

Note that if the above option is set in the database, then you will have to also create and set two values on the RDI3000 registry key. Open the Settings Editor and find the OpenEnterprise\Tasks\RDI3000 key. Then on it create the following values: -

Name	Description
EnableBSAUTO	Set to 1 to use BSAUTO to start OpenBSI. Used in conjunction with BSAUTOFormatString . When set to 0, the BSISStartSys API call will be used to start OpenBSI. Default value is 0.
BSAUTOFormatString	The format string used as the BSAUTO command line. Default value is "BSAUTO -std SYSTEM SYSTEM %s" where %s represents the NDF filename specified within the nw3000driver. obsistartndf attribute.

If RDI 3000 is running while this SQL statement is issued, the user will need to re-start the system for the change to take effect.

1.3.2 Detecting Devices Going Offline

To control the amount of time for a dead device to be detected, or to reduce network loading, set the following attributes within the nw3000Driver table: sanityperiod, sanityslice and maxiofails. The default values are 60 seconds, -1 and 3 respectively, which results in all devices being checked once every 60 seconds. After 3 consecutive failures, the device will be marked as offline.

The following SQL statement will result in a detection rate of approximately 60 seconds.

```
UPDATE nw3000driver SET Sanityperiod = 15, maxiofails = 1;
```

If RDI3000 is running while this SQL statement is issued, the user will need to re-start the system for the change to take effect.

How far the user can reduce the sanityperiod depends upon the number of devices within the Network 3000 network.

By default, device failure alarms will be generated whenever a device goes offline.

1.3.3 RDI3000 Forcelock Checking and Timeout

The RDI3000 periodically checks its internal wait packet queue, looking for RTU requests for which no response has been received. If any such requests are found, they are timed-out with a completion status of -1003.

An example EventHistory entry for such an action would look like this viewed from the SQL client:-

```

+-----+-----+
| description |
+-----+-----+
| 'Attempt to Update Value of Remote Signal SOUTH:TANK3.LEVEL. to 1.0 from 0.0' |
| 'Update Failed for Value of Remote Signal SOUTH:TANK3.LEVEL. with Status -1003' |
+-----+-----+
    
```

1.3.3.1 Configuration

The functionality is enabled by default and the internal wait packet timeout will be approximately 3 times the default RTU message timeout defined within OpenBSI. When the RDI starts it will display the internal wait packet timeout value in its Window.

The following value can optionally be created to either change or disable internal wait packet timeout functionality: Open the Settings Editor and find the openEnterprise\Tasks\RDI3000 key. Then create the value below.

Name	Data
WaitPacketTimeoutMultiplier	Set to 0 (zero) to disable internal wait packet timeout checking. A value greater than zero results in a wait packet timeout of (WaitPacketTimeoutMultiplier * default RTU message timeout) seconds. Default = 3

1.4 Time Zone Issues

This section deals with how the OpenEnterprise RDI3000 works with Bristol Babcock devices, OpenBSI and Time Zone issues.

1.4.1 Time Zones and Daylight Saving Time

By default, all timestamps received from a NW3000 device will be adjusted from local time to UTC before being passed to the database. Currently this only applies to remote alarm timestamps. RBE reports contain a timestamp but this is not propagated through to the real time database.

The NW3000 RDI adjusts timestamps based on the following rules.

- Timestamps from the NW3000 device will either be in daylight saving adjusted time or UTC (Coordinated Universal Time, or GMT). Only non UTC timestamps will be adjusted from local time to UTC. The device timestamp type is determined by reading the Open BSI configuration file BSBSAP.INI at start-up. Daylight Saving adjustment is only performed if Windows is configured for daylight saving adjustment.
- All time zone and daylight savings information is sourced from the database (nw3000Driver entry).

1.4.2 Disabling Timestamp Adjustment

Timestamp adjustment can be disabled by creating the following value on the `OpenEnterprise\Tasks\rdi3000` key, using the Settings Editor.

Name	Data
DisableTimestampAdjustment	A value of 1 (one) will disable timestamp adjustment. A value of 0 (zero), the default value, will enable timestamp adjustment.

NOTE: this key is optional and if it doesn't exist the default value will apply.

1.4.3 OpenBSI Time Synchronization

The default behaviour for Open BSI is to time synchronize NW3000 devices with local time. Note, if system local time is not adjusted for daylight savings then the devices will be synchronized with none daylight adjusted time.

It is recommended that NW3000 devices are synchronized with non daylight adjusted local time. However, the system should function correctly even if daylight saving adjusted local time is used. In the latter case there is a possibility that some remote alarm timestamps may be adjusted to UTC incorrectly during periods of daylight saving changes.

Open BSI time synchronization is configured within BSBSAP.INI. At start-up, RDI 3000 will read this file and only perform timestamp adjustment if Open BSI is not configured to use UTC.

BSBSAP.INI should reside in the standard Windows™ folder and has the following format.

```
[TIME]
```

```
utc=1
```

A value of 1 (one) for 'utc' will enable UTC time synchronization. A value of 0 (the default) will result in local time synchronization.

At start-up, RDI 3000 will read Windows daylight saving time bias. If set, it will assume that NW3000 devices are being synchronized to daylight saving adjusted local time. However, if Open BSI is configured for UTC then no daylight saving adjustment will be performed.

2 Index

A

ACCOL Version Mismatch Alarm Conditions 3
Alarm Summary Table 4
Assigning Timeclasses..... 8
Auto Starting OpenBSI Using RDI3000 11

B

BSAP Network 5

C

Collection By Templates 6
Configuring Post DBB 1
Creating Timeclasses..... 7

D

Database Build..... 1
Daylight Saving Time 13
Detecting Devices Going Offline 12
Device Failure Alarm Conditions..... 2
Disabling Timestamp Adjustment 14

L

Logical Alarm Template Collection 9

M

Manually Triggering Template Collection.... 9
*Modifying Timeclass Collection Frequencies*8

N

NW3000 Driver..... 10

O

OpenBSI Time Synchronization..... 14
Overview 1

P

Preventing Stress..... 5
Prevention 5
Primary Key Updates 5
Purging..... 4
PWRUP Remote Alarm..... 4

R

RBE Signals 10

RDI3000 Forcelock Checking..... 12

S

Setting..... 8

T

Template Collection..... 8
Time Zone Issues 13
Time Zones..... 13
Timeout..... 12
TPB Scripts..... 1

U

Using..... 4



NOTICE

Remote Automation Solutions ("RAS"), division of Emerson Process Management shall not be liable for technical or editorial errors in this manual or omissions from this manual. RAS MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THIS MANUAL AND, IN NO EVENT SHALL RAS BE LIABLE FOR ANY INCIDENTAL, PUNITIVE, SPECIAL OR CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PRODUCTION, LOSS OF PROFITS, LOSS OF REVENUE OR USE AND COSTS INCURRED INCLUDING WITHOUT LIMITATION FOR CAPITAL, FUEL AND POWER, AND CLAIMS OF THIRD PARTIES.

Bristol, Inc., Bristol Babcock Ltd, Bristol Canada, BBI SA de CV and the Flow Computer Division are wholly owned subsidiaries of Emerson Electric Co. doing business as Remote Automation Solutions ("RAS"), a division of Emerson Process Management. ROC, FloBoss, ROCLINK, Bristol, Bristol Babcock, ControlWave, TeleFlow and Helicoid are trademarks of RAS. AMS, PlantWeb and the PlantWeb logo are marks of Emerson Electric Co. The Emerson logo is a trademark and service mark of the Emerson Electric Co. All other trademarks are property of their respective owners.

The contents of this publication are presented for informational purposes only. While every effort has been made to ensure informational accuracy, they are not to be construed as warranties or guarantees, express or implied, regarding the products or services described herein or their use or applicability. RAS reserves the right to modify or improve the designs or specifications of such products at any time without notice. All sales are governed by RAS' terms and conditions which are available upon request.

RAS does not assume responsibility for the selection, use or maintenance of any product. Responsibility for proper selection, use and maintenance of any RAS product remains solely with the purchaser and end-user.

Engineered and supported by:

Remote Automation Solutions,

Bristol Babcock Ltd. Blackpole Road, Worcester, WR3 8YB, UK

Reg office: 2nd Floor, Accurist House, 44 Baker Street, London, W1U 7AL

Registered in England and Wales, Reg No. 2580226

VAT Reg No. GB 880 9839 67

Emerson Process Management
Remote Automation Solutions
1100 Buckingham St
Watertown, CT 06795
T 1 (860) 945 2200
F 1 (860) 945 2278
www.EmersonProcess.com/Remote
binfo@EmersonProcess.com

Emerson Process Management
Remote Automation Solutions
Blackpole Road
Worcester, WR3 8YB
T 44 (0) 1905 856848
F 44 (0) 1905 856930
www.EmersonProcess.com/Remote
oedsupport@EmersonProcess.com

