

Configuration Example

This ENOTE provides the user with an explanation of the various sections and data used in a typical file.

```
//=====//
//      Delivery.csv
//
// Customer      : Integrator Seattle
// Ultimate Destination : Big Bank
// Sales Order   : 00103400
// Driver Configuration : Modbus RTU
// Configured By : GFM
// Date         : 23 Mar 00
//
//=====//
```

Look for Relevant Project information here.

```
//
// Common Information
//
```

```
Bridge
Title
DCC030 CC00103400 V1.00a
```

This Title appears on the top line of the RUI editors. Use this to indicate the configuration version loaded, and the relevant customer/job.

```
//=====//
//
// Data Arrays
//
```

```
Data_Arrays
Data_Array_Name,      Data_Format,      Data_Array_Length
DA_AI_01,             UInt16,           200
DA_AO_01,             UInt16,           200
DA_DI_01,             Bit,              200
DA_DO_01,             Bit,              200
```

Declare "protocol neutral" data buffers (Data Arrays) for storage of data to be passed between protocols. Each Array needs to know what data format to store the data in.

```
//=====
//
// Client Side Connections
//
```

```
Connections
Port,      Baud,  Parity,  Data_Bits,  Stop_Bits,  Protocol,  Poll_Delay
P8,        9600,  None,    8,           1,           Modbus_RTU, 0.100s
```

Define the port to be connected to in terms of connection speed and properties.

Define the protocol for the network connected to this port.

Timing parameters on the connection allow for fine tuning of the communications.

```
//=====
//
// Client Side Nodes
//
```

```
Nodes
Node_Name, Node_ID, Protocol, Port
PLC 1,     1,       Modbus_RTU, P8
```

Allocate a name to the node for reference by the map descriptors.

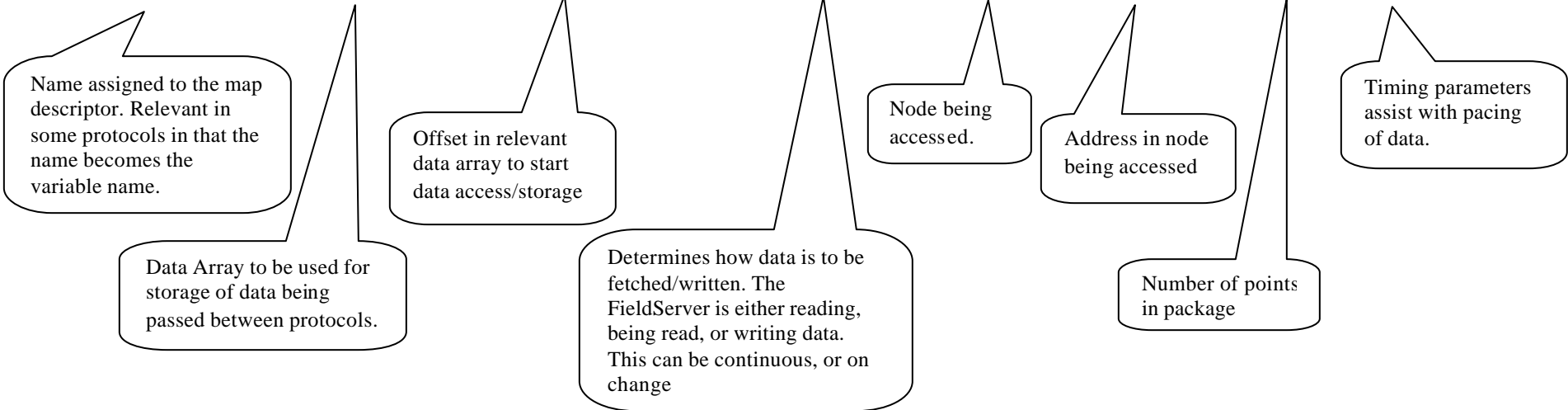
This is the Node ID of the server. If the FieldServer is a server, then this represents the FieldServer Node ID.

If the FieldServer is a client, then the remote server node is attached to the connection. If the FieldServer is a server, then the node can be accessed via any valid port declared in "Connections", and this parameter is not declared in the node.

```
//=====
//
// Client Side Map Descriptors
//
```

Map_Descriptors							
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name,	Address,	Length,	Scan_Interval
CMD_AI_01,	DA_AI_01,	0,	RDBC,	PLC 1,	30001,	20,	1.000s
CMD_AO_01,	DA_AO_01,	0,	WRBC,	PLC 1,	40001,	20,	1.000s

Map_Descriptors							
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name,	Address,	Length,	Scan_Interval
CMD_DI_01,	DA_DI_01,	0,	RDBC,	PLC 1,	10001,	20,	1.000s
CMD_DO_01,	DA_DO_01,	0,	WRBX,	PLC 1,	00001,	20,	1.000s



```
//=====
//
// Server Side Connections
//
```

```
Connections
Adapter,      Protocol
N1,          Modbus/TCP
```

Adapter definition applies to defining network and hardware protocol (eg. Profibus) connections.

Define the protocol for the network connected to this port.

```
//=====
//
// Server Side Nodes
//
```

```
Nodes
Node_Name,   Node_ID,   Protocol
MBP_Srv_11,  11,          Modbus/TCP
```

Allocate a name to the node for reference by the map descriptors.

This is the Node ID of the server. Since the bridge is a server here, this represents the bridge (virtual) Node ID. It is possible for the bridge to represent multiple Virtual Node ID's in most protocols.

```
//-----
//
// Server Side Map Descriptors
//
```

Map_Descriptors

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name,	Address,	Length,	Data_Array_Low_Scale,	Data_Array_High_Scale,	Node_Low_Scale,	Node_High_Scale
SMD_AI_01,	DA_AI_01,	0,	Server,	MBP_Srv_11	30001,	200,	0,	100,	32,	212
SMD_AO_01	DA_AO_01,	0,	Server,	MBP_Srv_11	40001,	200,	0,	100,	32,	212

Scaling.
The last four parameters shown here can be used to scale data array values before passing them onto the node. The way this should be read is: A value from 0 to 100 in the data array represents a value from 32 to 212 on the node (This example converts Celcius to Fahrenheit). This is not a range, and so a value of 500 (for example) will be correctly transferred.

Map_Descriptors

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name,	Address,	Length
SMD_DI_01,	DA_DI_01,	0,	Server,	MBP_Srv_11,	10001,	200
SMD_DO_01,	DA_DO_01,	0,	Server,	MBP_Srv_11,	00001,	200

Name assigned to the map descriptor. Relevant in some protocols in that the name becomes the variable name.

Data Array to be used for storage of data being passed between protocols.

Offset in relevant data array to start data access/storage

Determines how data is to be fetched/written. The bridge is either reading, being read, or writing data. This can be continuous, or on change

Bridge node to which the data is being allocated.

Address in bridge node to which the referenced data in the data array is being allocated.

Number of points in package