



## C-Gate Server Application

### What's New

5000 CG Series



## Table of Contents

<b>Section</b>	<b>Page</b>
1.0 New Features in C-Gate 1.5	3
1.1 Introduction	3
1.2 Unit Address 255 is now legal	3
1.3 Multiple Applications	4
1.4 Addressing Changes	5
1.5 Access Control	7
2.0 Better ramp support	10
2.2 Change Unit Addresses	10
2.3 Unravel support	10
2.4 C-Bus Bridge 1.0 Support	11
2.5 Load Change Port	12
2.6 Configuration Change Port	12
2.7 SceneMaster Unit	13
2.8 CI Command supports Multiple applications and ramp time store	13
2.9 Advisory object locking	13

### Copyright Notice

© Copyright 2002 Clipsal Integrated Systems Pty Ltd. All rights reserved

### Trademarks

- Clipsal is a registered trademark of Gerard Industries Pty Ltd.
  - C-Bus is a registered trademark of Clipsal Integrated Systems Pty Ltd.
  - Intelligent Building Series is a registered trademark of Clipsal Integrated Systems Pty Ltd.
- All other logos and trademarks are the property of their respective owners.

### Disclaimer

Clipsal Integrated Systems reserves the right to change specifications or designs described in this manual without notice and without obligation.

## 1.0 New Features in C-Gate 1.5

This document describes the new features in Version 1.5 of C-Gate.

### 1.1 Introduction

C-Gate 1.5 introduces a series of new features and changes. This guide gives an overview of the new features and important changes.

Major changes in this version are:

1. Unit address \$FF or 255 is now a legal unit address
2. C-Gate 1.5 supports **multiple C-Bus applications per network**. This means that the way that groups and applications are addressed has changed.
3. Major changes to the way **addresses** are given for objects in C-Gate and on the C-Gate networks, including access to wildcards to select multiple objects for one command.
4. **Access control** is provided, by the new **login** command and the **access.txt** configuration file. Access control works on a per-command and per login session basis. Also, additional connection control is provided.
5. **Better Ramp support** to provide other applications ramp information, and to model ramp operations in C-Gate as they occur.
6. **Unit Addresses can be changed** from the command line.
7. C-Gate can comprehensively **unravel** a network that has overlapping unit addresses.
8. C-Gate supports access to networks using **C-Bus Bridge** Units.
9. C-Gate supports the SceneMaster unit.
10. CI command supports multiple applications and ramp time store
11. Advisory locking provided for programming software
12. C-Bus Bridge 2.0 is supported.
13. Software-based licensing scheme is available as an alternative to the hardware key approach.

See below for details of these changes.

### 1.2 Unit Address 255 is now legal

A device with a unit address of \$ff or 255 is now a legal unit address. Prior to 1.5, unit address \$ff of 255 was considered an illegal unit address and as a result the unit would not operate correctly.

### 1.3 Multiple Applications

C-Gate 1.5 supports more than one application per network. Previously, C-Gate allowed one application for the entire project. This was the default application that was set in the config parameter `C-Bus-application` and was set to 56 (\$38) as a default. C-Gate ignored other applications and events from the network concerning other applications.

As of version 1.5, C-Gate supports all the applications that are defined in a network. At initial network scan time, C-Gate checks for all valid applications and includes them in that network. These applications are then valid for this network.

C-Gate checks both the primary and secondary application for each unit to establish the applications in use. The primary application is obtained using the `get <unit> application` command, while the secondary can be obtained with `get <unit> application2` command.

This changes the hierarchy of objects in C-Gate. In previous versions, A C-Bus network has a set of groups associated with it. That set of groups were groups in the default application. Now, a C-Gate network has a set of applications associated with it, and each application has a set of groups associated with it.

This results in changes in addressing.

Whereas before, you would enter the command:

```
on 1.1
```

to turn on group 1 of the default application of network 1, you now enter:

```
on 1/$38/1
```

or

```
on 1/56/1
```

See the next section on the new addressing to get a feel for the way addressing now works. Similarly, output from events now uses the `'/'` addressing syntax, so that application information is included.

Note that the existing address style `'1.1'` can still be used, and that this then uses the default application for the particular network.

### 1.3.1 Default Application

Each network still has a default application. This is initially derived from the `cbus-application` startup parameter but can be set on an individual network basis by using the `DefaultNetwork` parameter that can either be set from the command line or from a network startup file.

## 1.4 Addressing Changes

There are big changes to the way that addresses of C-Bus groups, networks, applications etc are given.

Existing addressing is still supported, but is somewhat limited, whereas new addressing allows addressing of multiple applications.

These addresses are used:

- wherever a C-Bus group address is used in an on, off, or ramp command
- wherever a network, unit, terminal, or application is addresses from a get or show, set, do, recall, getstatus, getstate, tree or other command

### 1.4.1 New Addresses

To specify a C-Bus group, the form of the address is:

```
/network/application/group
```

where the / symbol is used to separate elements of the address, and the bold words are replaced with address parts. A leading / is currently optional, but **may be used later to include project naming, so indiscriminate use is discouraged.**

For example:

```
5/56/1
```

or

```
5/$38/1
```

would represent group 1, application 56 (hex 38), on C-Bus network 5. Note that a hexadecimal value can be given for a numeric network part by preceeding it with a \$ character.

You can also name a group in the default application by using either of the following forms:

5//1

or

5/~ /1

where the double slash // or the ~ indicate the default application **for that network** is to be used.

For physical network components (units and terminals), a component is added to the path. To refer to a unit, either of the following forms are used:

network/p/unit

or

p/network/unit

the **p** component of the address indicates that a unit is being addressed. Similarly, an individual dimmer terminal can be specified as:

network/p/unit/terminal

or

p/network/unit/terminal

For example:

p/5/11/1

refers to terminal or circuit 1 of unit 11 on network 5.

A '\$' character before a numeric address part renders the following number as though it were hexadecimal. Note that network names are not necessarily numeric, so this does not apply to the network portion of an address.

For example the address for the generic lighting application on network 1 is:

1/\$38

This is equivalent to

1/56

### 1.4.2 Output using new addresses

All output from C-Gate now uses these new addresses, including:

- results from get and set and other command
- event output

This will require changes in applications that scan responses for addresses.

### 1.4.3 Wildcard addressing

You can use a \* character as a wildcard in one or more address parts, meaning that the command will be applied to all the matching objects.

For example, you can execute the command

```
on 1/56/*
```

to turn on all the groups in application 56 on network 1. Use this carefully as it will turn on everything.

You will see one result for each object that the command is executed on, or a failure if applicable.

Note you can apply this to any command with an address, including get and set operations, recall, getstatus, store and tree.

### 1.4.4 1.0-style addressing

For legacy applications reading the command line or event stream, it is possible to make C-Gate report addresses in the old '.' separated address syntax. Setting the config file parameter **use-1.0-addressing** to **yes** will enable this mode. Note that addresses for other than the default application are still given using the new address syntax.

## 1.5 Access Control

C-Gate 1.5 has a stronger focus on access control, which is controlled by the **access.txt** file in the **config** directory (where the Networks.txt and CGateConfig.txt files reside).

This file has a list of types of access based on three keywords:

- interface defines an access level for a TCP/IP interface on the computer
- remote defines an access level for a remote connecting IP address
- user defines a user & password and access level set by the new login command

The access control file is checked every time a connection is made to C-Gate command event or status port, or a **login** command is received. At these times, the list of rules in the config file are reviewed, and the access level granted is the result of the most favourable rule. However, using the login command, it is

possible to set your access level to a lower level than would normally be configured by the interface and remote rules.

Access levels are as follows, with each later level incorporating the functions of the previous level.

Level	Meaning
<b>none</b>	No access at all. Use this to refuse connections.
<b>connect</b>	allow a connection to be established (to the command interface only) and execute the login command
<b>monitor</b>	Allow monitoring and query of status of objects and C-Bus, but do not allow any changes. This is the minimum level to allow connection to the event or load or configuration change ports.
<b>operate</b>	allow set, on, off, ramp operations – allow changes to be made to the system
<b>admin</b>	allow C-Gate shutdown and C-Gate administration functions
<b>program</b>	allow C-Bus networks to be programmed
<b>debug</b>	allow debugging functions to be performed

The formation rules for the arguments are as follows:

**interface** <IP address> <access level>

**remote** <IP address> <access level>

**user** <userid> <password> <access level>

Where the fields are:

**<IP address>** a full IP address, either for the interface that the connection comes in on, or a remote address. This can be a hostname, or it can be a dotted quad (ie. 10.234.2.23). If a dotted quad is used, then any part of the address that is set to 255 will match any address, allowing whole networks to be allowed access.

**<userid>** a username that is used in the login command along with the password.

**<password>** the password that goes with the username.

### 1.5.1 A sample access control file

A sample access control file would be:

```
#sample access control file
interface 127.0.0.1 admin
interface 10.50.1.2 monitor
remote 10.50.1.255 operate
user debugger xxAA1 debug
```

This file would set access control in the following way:

1. All sessions from the local computer on the localhost interface would get admin level access.
2. All sessions on the Ethernet interface 10.50.1.2 (i.e. all access from the external network) would get monitor level
3. All sessions on the 10.50.1.\* sub network would get operate access
4. A user that typed the command:  
login debugger xxAA1  
  
would be granted debug access level.

Note that the access.txt file is scanned every time a connection is made or a login command is entered, so changes can be made to it and the new rules will apply without rebooting, though they won't change the behaviour of existing connections, unless a login command is executed).

## 2.0 Better ramp support

C-Gate's handling of ramp commands has been greatly improved.

### 2.1.1 Reporting ramp commands:

C-Gate now reports a ramp time as well as a level in the event stream. When a ramp is seen on a network, it is now reported as:

```
20010608-173823.708 730 2/57/1 new level=255 sourceunit=2 ramp time=180
```

The ramp time parameter given is the time taken for a minimum through maximum ramp. On and off commands are shown with a ramp time of zero.

### 2.1.2 Ramp modeling performed

When a group or terminal is ramping, C-Gate returns a calculated ramp level in response to a query for a level. This prevents jumping and confusion in user interfaces.

## 2.2 Change Unit Addresses

C-Gate now supports the changing of the Unit Address of a C-bus Unit. This is done using the `set <unit> address <value>` operation. This returns a 200 OK if the operation was successful, and will show the unit updated in the model.

## 2.3 Unravel support

A network can be unraveled using the `do <network> unravel` method.

This will move through all the units on the network and look for units with duplicate addresses, relocating units are required to locate duplicates.

**Warning:** running unravel will alter the unit addresses of your network. At the very least, the unravel process will:

- move the address of the PCI to an intermediate address, and then back to its initial address, unless it is at address 1, in which case it is not moved.
- move the location of any bridges that C-Gate is communicating through, and then move them back at the conclusion of the unravel process.

## 2.4 C-Bus Bridge 1.0 Support

C-Gate supports a new network type, called **bridge** that allows access across C-Bus networks through C-Bus Bridge units. To define a bridged network, add an entry to the Networks.txt file in the form:

```
<name> bridge <bridge-address>
```

where:

```
<name> is the network name  
<bridge-address> is the C-Gate address for the bridge that is used  
to get to this new network.
```

For example:

```
9 bridge p/1/9
```

defines a new network **9** that is attached to network 1 via the bridge at unit address 9.

C-Gate looks for bridged networks once the host network has been scanned, so bridged networks are scanned after directly-connected networks.

Limitations:

- Monitoring is not available. Limited monitoring that works for a **single level of bridging only** is available as long as the Bridge is set up to send application messages from the bridged network to the directly connected network.
- Unlike a directly connected network, the bridge does not provide positive or negative feedback as to the successful transmission of a command. Consequently, messages may be lost and go unnoticed until the next sync operation takes place. **C-Bus Bridge version 2.0** provides enhanced reliability in this respect.

If C-Gate detects a C-Bus 2.0 bridge, it places the bridge into monitor mode, so all events on that network are available for monitoring.

## 2.5 Load Change Port

The load change port provides a list of on, off and ramp commands as they occur on networks that C-Gate has open. A client connects to port 20025 (or the port set by the `load-change-port` configuration parameter) and then as load changes (changes to C-Bus Groups) occur on the networks that C-Gate is managing, output in the form of a series of on, off and ramp commands is provided. A sample output is:

```
ramp 1/57/22 0 300
off 1/57/0
on 9/57/0
off 9/57/2
off 9/57/41
```

Any number of clients can connect to this port.

Connecting clients must have at least `monitor` level access control to complete the connection.

## 2.6 Configuration Change Port

The load change port provides a list of device addresses (units or groups) in which programming or reconfiguration has occurred as the reconfiguration occurs.. A client connects to port 20026 (or the port set by the `config-change-port` configuration parameter) and then as configuration changes occur on the networks that C-Gate is managing, output in the form of a series of object addresses is provided.

The configuration events that are captured are:

- A change in unit address for a unit – this causes an event from the affected network
- A store command sent to any unit (any change in unit parameters)
- The initial creation of a unit (via a new command or a network scan)

A sample output from this port is:

```
6
p/6/165
p/6/2
p/6/2
6
p/6/2
p/6/165
p/6/165
```

This output is from an unravel operation on network 6. What it shows is the movement of unit p/6/165 to new address p/6/2 and back again.

Connecting clients must have at least `monitor` level access control to complete the connection.

## **2.7 SceneMaster Unit**

C-Gate models the SceneMaster unit (type SCNCTL5) to a limited extent. It captures all of the groups in use by the SceneMaster, but it does **not** get target levels from the unit.

## **2.8 CI Command supports Multiple applications and ramp time store**

The CI command supports multiple applications by giving an **application** address (in the form <network>/<application>) in place of a network address in the CI command. Specifying only a network results in access to the groups of the default application for that network.

The CI command supports a new data type, 16, which stores a ramp time value (in seconds) to a C-Bus group. This allows a CI group setting command to execute a timed ramp, where the ramp time value has been store previously.

## **2.9 Advisory object locking**

An advisory locking system has been introduced to allow programming application to coordinate changes to objects. One lock is available per object (network, unit, etc) and this lock is held by a command session. A session gets a lock on an object using the `lock <object>` command and unlocks it with the `unlock <object>` command.

This locking is only advisory: The lock does not enable or prevent other C-Gate commands from operating. It is up to applications using the locks to manage their changes and use of locks together.