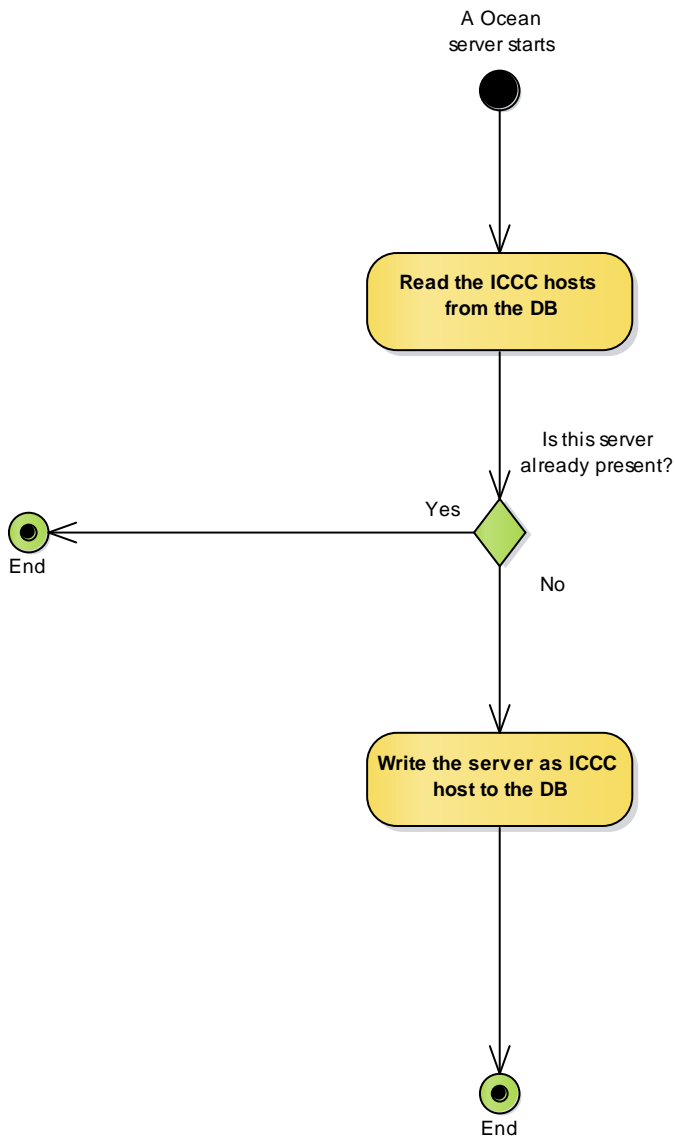# Ocean's ICCC Terminologies

Listener := An ICCC command at a given channel which is located on any or all of the servers
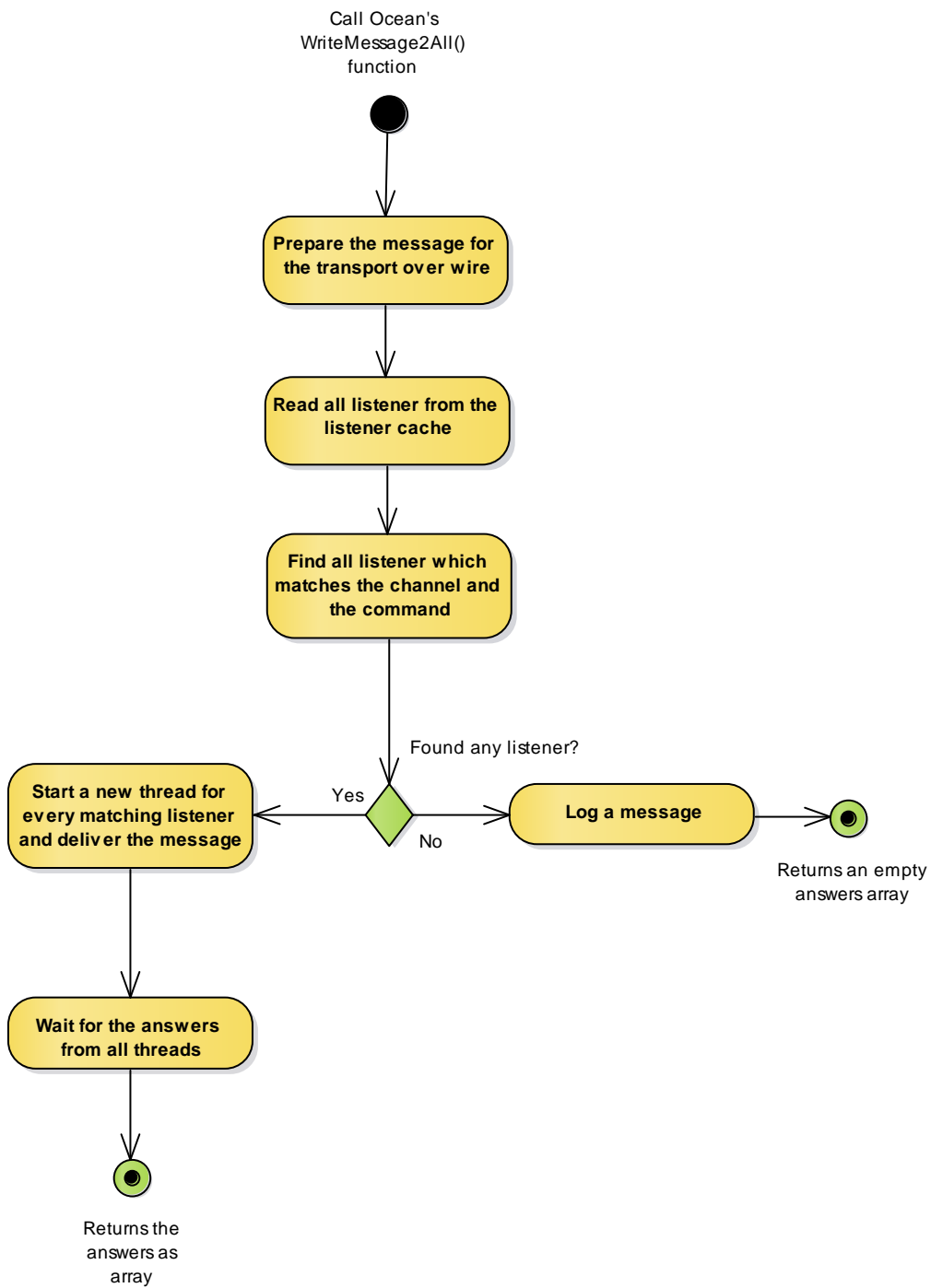
Host := An ICCC host is a server which provides at least one ICCC command. Thus, any host provides at least one listener.

Ocean's ICCC start-up

A Ocean
server starts

●

**Read the ICCC hosts
from the DB**

Is this server
already present?

Yes ◆ No

End ◉

**Write the server as ICCC
host to the DB**

End ◉

This diagram shows only the ICCC parts
of Ocean's start-up process! Further, this
diagram is only for Ocean true -- other
external ICCC components e.g. with
C#/.NET, Python, etc. are setting up
differently.

Ocean's WriteMessage2All() function

Call Ocean's
WriteMessage2All()
function

●

Prepare the message for the transport over wire

Read all listener from the listener cache

Find all listener which matches the channel and the command

Found any listener?

Yes ◆ No

Start a new thread for every matching listener and deliver the message

Log a message

◉

Returns an empty
answers array

Wait for the answers from all threads

◉

Returns the
answers as
array

Ocean's WriteMessage2Any() function

Call Ocean's
WriteMessage2Any()
function

Prepare the message for the transport over wire

Read all listener from the listener cache

Find all listener which matches the channel and the command

Found any listener?

Yes

No

Choose a random listener

Log a message

Returns an empty answer

Delivers the prepared message

Wait for the answer

Returns the answer

Requirement: Each external host is enabled to send directly messages to any other listener

«use»

Requirement: External ICCC hosts must be able to read the listeners and hosts from the MongoDB cluster

«instantiate»

Risk: Through the ICCC drivers, any third-party developer has access to Ocean's MongoDB cluster

«derive»

Requirement: External ICCC hosts have no direct access to Ocean's MongoDB cluster.

«serve»

Risk: The list of hosts can grow fast for IoT/CPS or general for virtual and cloud environments. After the shutdown of such systems, many dead hosts are listed!

«derive»

Requirement: Any host should be able to perform healthy checks against other hosts.

«serve»

«serve»

Feature: Built-in ICCC command for healthy checks i.e. a ping command

Feature: Built-in ICCC command to delete a dead host

Feature: Through the ICCC, the external driver can ask any Ocean server to get the currently known listeners and hosts

«serve»

«derive»

Requirement: To set-up a ICCC driver, some or all addresses of the Ocean servers must be known

«serve»

Feature: The user of the ICCC driver must be able to provide Ocean server's addresses

Requirement: The ICCC driver must maintain the own listeners

«serve»

«serve»

Feature: The ICCC driver is able to enable or disable his own listeners

Feature: The ICCC driver is able to remove i.e. delete it's own listeners

«derive»

«derive»

Requirement: Ocean must support the external drivers

«serve»

Feature: Built-in ICCC command to enable or disable a listener

«derive»

«serve»

«serve»

Feature: Built-in ICCC command to delete an external listener

Feature: Built-in ICCC commands for getting the listeners and hosts

Class diagram for an Ocean ICCC driver

ICCC
- + ICCCConnection
- + ICCCHost
- + ICCCListener
- + ICCCProcessor
- + IICCCMessage
- + SystemMessages

ICCC::ICCCConnection
- icccHostsCache: IEnumerable<ICCCHost>
- icccListenerCache: IEnumerable<ICCCListener>
- + INSTANCE: ICCCConnection {readOnly}

- + send2All(IICCCMessage): IEnumerable<IICCCMessage>
- + send2Any(IICCCMessage): IICCCMessage
- «constructor»
- - ICCCConnection()

SystemMessages
- + ICCCComponentStartUpMessage
- + ICCCDefaultAnswer
- + ICCCDeleteHost
- + ICCCDeleteListener
- + ICCCGetHosts
- + ICCCGetHostsAnswer
- + ICCCGetListeners
- + ICCCGetListenersAnswer
- + ICCCListenerUpdate
- + ICCCNumGenNext
- + ICCCNumGenNextAnswer
- + ICCCOceanStartUpMessage
- + ICCCPing
- + ICCCRegisterHost
- + ICCCRegisterListener

(from ICCC)

Channel: System::Startup
Command: System::ComponentStart
Answer: Default

Channel: System::Startup
Command: System::OceanStart
Answer: Default

Channel: System::ICCC
Command: ICCC::RegisterListener
Answer: Default

ICCC::ICCCListener
- channel: string
- command: string
- ipAddressPort: string
- isActive: bool

- «property»
- + Channel(): string
- + Command(): string
- + IPAddressPort(): string
- + IsActive(): bool
- «constructor»
- + ICCCListener()
- + ICCCListener(string, string, string, bool)

SystemMessages::ICCCComponentStartUpMessage
- ipAddressPort: string
- name: string
- version: string

- «constructor»
- + ICCCComponentStartUpMessage()
- «property»
- + IPAddressPort(): string
- + Name(): string
- + Version(): string

SystemMessages::ICCCOceanStartUpMessage
- adminIPAddressPort: string
- oceanVersion: string
- publicIPAddressPort: string

- «property»
- + AdminIPAddressPort(): string
- + OceanVersion(): string
- + PublicIPAddressPort(): string
- «constructor»
- + ICCCOceanStartUpMessage()

SystemMessages::ICCCRegisterListener
- «constructor»
- + ICCCRegisterListener()

Channel: System::ICCC
Command: ICCC::DeleteHost
Answer: Default

Channel: System::ICCC
Command: ICCC::ListenerUpdate
Answer: Default

Channel: System::Ping
Command: Ping::Ping
Answer: Deafult

SystemMessages::ICCCPing
- «constructor»
- + ICCCPing()

SystemMessages::ICCCDeleteHost
- hostname: string
- ipAddressPort: string

- «property»
- + Hostname(): string
- + IPAddressPort(): string
- «constructor»
- + ICCCDeleteHost()

SystemMessages::ICCCListenerUpdate
- channel: string
- command: string
- ipAddressPort: string
- isActiveNew: bool

- «property»
- + Channel(): string
- + Command(): string
- + IPAddressPort(): string
- + IsActiveNew(): bool
- «constructor»
- + ICCCListenerUpdate()

ICCC::ICCCHost
- hostname: string
- ipAddressPort: string

- «property»
- + Hostname(): string
- + IPAddressPort(): string
- «constructor»
- + ICCCHost()
- + ICCCHost(string, string)

Channel: System::NumGen
Command: NumGen::Next
Answer: ICCCNumGenNextAnswer

SystemMessages::ICCCNumGenNext
- «constructor»
- + ICCCNumGenNext()

«interface»
ICCC::IICCCMessage

SystemMessages::ICCCRegisterHost
- «constructor»
- + ICCCRegisterHost()

Channel: System::ICCC
Command: ICCC::RegisterHost
Answer: Default

SystemMessages::ICCCNumGenNextAnswer
- number: long

- «constructor»
- + ICCCNumGenNextAnswer()
- «property»
- + Number(): long

Channel: System::ICCC
Command: ICCC::GetListeners
Answer: ICCCGetListenersAnswer

SystemMessages::ICCCGetListeners
- «constructor»
- + ICCCGetListeners()

SystemMessages::ICCCGetHostsAnswer
- addressesPorts: string[]
- hostnames: string[]

- «property»
- + Hostnames(): string[]
- + IPAddressesPorts(): string[]
- «constructor»
- + ICCCGetHostsAnswer()

SystemMessages::ICCCGetHosts
- «constructor»
- + ICCCGetHosts()

SystemMessages::ICCCDefaultAnswer
- + AnswerACK: ICCCDefaultAnswer = true, 0 {readOnly}
- + AnswerNACK: ICCCDefaultAnswer = false, 0 {readOnly}
- + AnswerUNKNOWN: ICCCDefaultAnswer = false, -1 {readOnly}
- commandAnswer: long
- commandSuccessful: bool

- «property»
- + CommandAnswer(): long
- + CommandSuccessful(): bool
- «constructor»
- + ICCCDefaultAnswer()
- + ICCCDefaultAnswer(long, bool)

SystemMessages::ICCCDeleteListener
- channel: string
- command: string
- ipAddressPort: string

- «property»
- + Channel(): string
- + Command(): string
- + IPAddressPort(): string
- «constructor»
- + ICCCDeleteListener()

SystemMessages::ICCCGetListenersAnswer
- addressesPorts: string[]
- channels: string[]
- commands: string[]

- «property»
- + Channels(): string[]
- + Commands(): string[]
- + IPAddressesPorts(): string[]
- «constructor»
- + ICCCGetListenersAnswer()

Channel: System::ICCC
Command: ICCC::GetHosts
Answer: ICCCGetHostsAnswer

Channel: System::ICCC
Command: ICCC::DeleteListener
Answer: Default

ICCC::ICCCProcessor
- + INSTANCE: ICCCProcessor {readOnly}

- ~ convertData2Message(NameValueCollection): IICCCMessage
- ~ convertMessage2Data(IICCCMessage): NameValueCollection
- «constructor»
- - ICCCProcessor()

Start-up process for a external ICCC driver / host

An external
ICCC host
starts

Read the given parameters of the caller

Was at least one Ocean server address provided?

No → **Use 127.0.0.1:60000**

Yes → **In case of several addresses: Choose a random server**

Request the current listeners and hosts

Was the request successful?

No → **Re-try the request**

Yes → **Henceforth, use the current host list for further requests**

Start a maintenance thread which updates the listener and host list every five minutes

Set-up and send the own start-up message to any sever in order to allow others to call us

Was the request successful?

No → **Re-try the request**

Yes → Done

Documentation of ICCC Messages

Message: ICCCGetListeners

Channel: System::ICCC
Command: ICCC::GetListeners
Answer: ICCCGetListenersAnswer

Behavior Ocean internal:
... Send: Never, will use internal function
... Receive: Provides all known listeners

Behavior external components:
... Send: To obtain all listeners
... Receive: Never
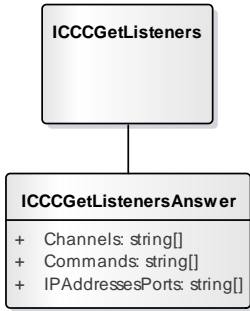
ICCCGetListeners

ICCCGetListenersAnswer

+ Channels: string[]
+ Commands: string[]
+ IPAddressesPorts: string[]

Message: ICCCRegisterHost

Channel: System::ICCC
Command: ICCC::RegisterHost
Answer: Default

Behavior Ocean internal:
... Send: Never, will use internal function
... Receive: Registers the host

Behavior external components:
... Send: To register own host
... Receive: Never

ICCCRegisterHost

+ Hostname: string
+ IPAddressPort: string

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool
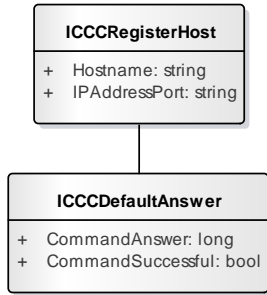
Message: ICCCGetVersion

Channel: System
Command: System::Version
Answer: ICCCGetVersionAnswer

Behavior Ocean internal:
... Send: On admin's request; Kind=Ocean; Name=Hostname
... Receive: Reports own version

Behavior external components:
... Send: On admin's request; Kind=Component; Name=Component's name
... Receive: Reports own version & name

ICCCGetVersion

ICCCGetVersionAnswer

+ Kind: string
+ Name: string
+ Version: string
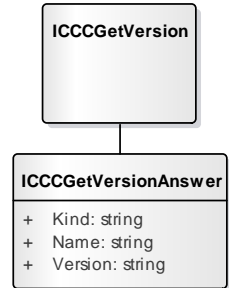
Message: ICCCComponentStartUpMessage

Channel: System::Startup
Command: System::ComponentStart
Answer: Default

Behavior Ocean internal:
... Send: Never
... Receive: Log the content

Behavior external components:
... Send: After start-up
... Receive: Logging is possible

ICCCComponentStartUpMessage

+ IPAddressPort: string
+ Name: string
+ Version: string

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool
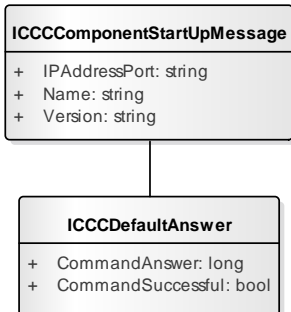
Message: ICCCOceanStartUpMessage

Channel: System::Startup
Command: System::OceanStart
Answer: Default

Behavior Ocean internal:
... Send: After start-up of any Ocean server
... Receive: Log the content

Behavior external components:
... Send: Never
... Receive: Logging is possible
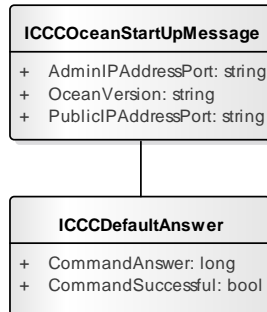
ICCCOceanStartUpMessage

+ AdminIPAddressPort: string
+ OceanVersion: string
+ PublicIPAddressPort: string

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool

Message: ICCCRegisterListener

Channel: System::ICCC
Command: ICCC::RegisterListener
Answer: Default

Behavior Ocean internal:
... Send: Never, will use internal function
... Receive: Register the listener

Behavior external components:
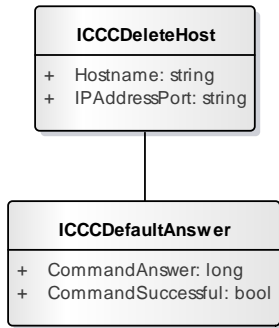... Send: Anytime to register own listeners.
... Receive: Never

ICCCRegisterListener

+ Channel: string
+ Command: string
+ IPAddressPort: string
+ IsActive: bool

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool
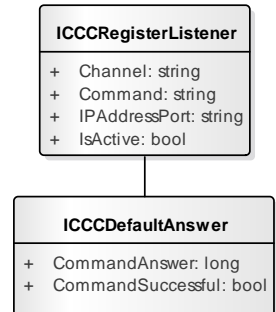
Message: ICCCDeleteHost

Channel: System::ICCC
Command: ICCC::DeleteHost
Answer: Default

Behavior Ocean internal:
... Send: To delete a dead host entry
... Receive: Delete the host

Behavior external components:
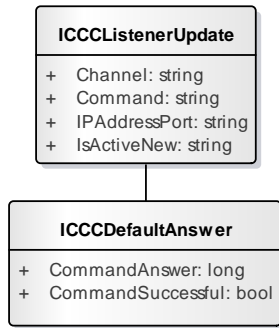... Send: To delete own or dead host
... Receive: Never

ICCCDeleteHost

+ Hostname: string
+ IPAddressPort: string

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool

Message: ICCCListenerUpdate

Channel: System::ICCC
Command: ICCC::ListenerUpdate
Answer: Default

Behavior Ocean internal:
... Send: Never
... Receive: Update the IsActive status for this listener

Behavior external components:
... Send: To update known listeners
... Receive: Never

ICCCListenerUpdate

+ Channel: string
+ Command: string
+ IPAddressPort: string
+ IsActiveNew: string

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool

Message: ICCCNumGenNext

Channel: System::NumGen
Command: NumGen::Next
Answer: ICCCNumGenNextAnswer

Behavior Ocean internal:
... Send: Never, will use internal function
... Receive: Delivers a new unique number

Behavior external components:
... Send: To obtain a unique number
... Receive: Never

ICCCNumGenNext

ICCCNumGenNextAnswer

+ Number: long
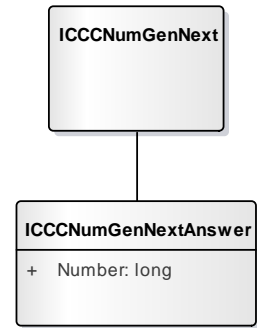
Message: ICCCGetHosts

Channel: System::ICCC
Command: ICCC::GetHosts
Answer: ICCCGetHostsAnswer

Behavior Ocean internal:
... Send: Never, will use internal function
... Receive: Delivers current known hosts

Behavior external components:
... Send: To obtain all known hosts
... Receive: Never

ICCCGetHosts

ICCCGetHostsAnswer

+ Hostnames: string[]
+ IPAddressesPorts: string[]
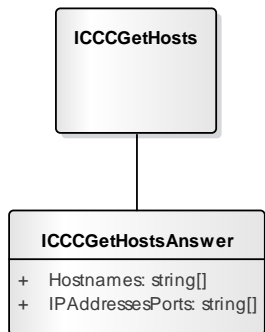
Message: ICCCPing

Channel: System::Ping
Command: Ping::Ping
Answer: Deafult

Behavior Ocean internal:
... Send: To check the status of other hosts
... Receive: Assume the host as healthy

Behavior external components:
... Send: To check the status of other hosts
... Receive: Assume the host as healthy
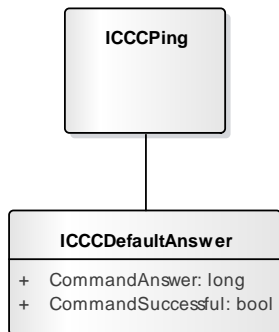
ICCCPing

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool

Message: ICCCDeleteListener

Channel: System::ICCC
Command: ICCC::DeleteListener
Answer: Default

Behavior Ocean internal:
... Send: Never, will use internal function
... Receive: Deletes the listener

Behavior external components:
... Send: To maintain listeners
... Receive: Never

ICCCDeleteListener

+ Channel: string
+ Command: string
+ IPAddressPort: string

ICCCDefaultAnswer

+ CommandAnswer: long
+ CommandSuccessful: bool