Created: November 18, 1996
Modified : August 14, 1997 Saved: November 20, 1997

1

Gemini Controls Group Interface Control Document

ICD 7a — ICS Subsystem Interfaces

Steven Beard, Steve Wampler and Chris Mayer

gscg.grp.015-ICD07a/20 cics_smb_014/20

This document defines the Instrument Control System subsystem interfaces that need consistent implementation with other subsystem interfaces.

CONTENTS

1.0 Introduction

1.1 Purpose

An overview describing all the interfaces in the Gemini system is given in document "gscg.grp.005, Gemini System Interfaces, Gemini 8m Telescopes Project." (which is reference [3] as listed on page 4). A description of the guidelines to be followed by Gemini Interface Control Documents may be found in reference [4].

This Interface Control Document (ICD) specifies the general way in which a Gemini Instrument Control System will communicate with its subsystems. It specifies those interfaces in the Gemini system which, while most of them exist entirely within the ICS, share a common implementation with similar interfaces found in the TCS and described in ICD/7b [11]. As such, these interfaces should be as consistent as is reasonable between the ICS and TCS to reduce development, maintenance, acceptance, and upgrade costs.

The interfaces involved are as follows:

1.1.1 Internal control interfaces

1.1.1.1 Interfaces governed by this ICD

- Control commands and responses from the ICS 'Instrument Sequencer' to the Detector Controller and Components Controller for a science instrument.
- Control commands and responses from any instrument-like TCS subsystem to any Detector Controllers and Components Controllers under its control (for example the interface

between the A&G GBD sequencer and the Detector Controllers and Components Controllers for its peripheral wavefront sensors — see [25]).

- Control commands and responses from the A&G GBD sequencer to the Components Controller of an on-instrument wavefront sensor (OIWFS) installed in the IOC of an instrument. The interface is logically exactly the same as that between an Instrument Sequencer and Components Controller, except in this case the two systems happen to be on different IOCs (see [25] and [10]).

1.1.1.2 Related interfaces governed by ICD/7b, [11]

- Control commands and responses from the TCS Master Control System to all TCS subsystems (PCS, SCS, AO, A&G, MCS, CRS, and ECS).

1.1.2 Internal status interfaces

1.1.2.1 Interfaces governed by this ICD

- Status information from the ICS Detector Controller and Components Controller to the ICS ‘Instrument Sequencer’.
- Status information from the Detector Controllers and Components Controllers belonging to any instrument-like TCS subsystems to their master controller (for example status information from the peripheral wavefront sensor controllers to the A&G control system — see [25]).
- Status information from the OIWFS Components Controller installed in the IOC of an instrument to the A&G Sequencer (see [25] and [10]).

1.1.2.2 Related interfaces governed by ICD/7b, [11]

- Status information from the TCS subsystems to the TCS Master Control System.

1.1.3 Internal alarm interfaces

Note that alarms raised on this interface will almost certainly also be detected by the OCS on the ICD/2 Status/Alarms interface (see [7]).

1.1.3.1 Interfaces governed by this ICD

- Alarms from the ICS Detector Controller and Components Controller to the ICS ‘Instrument Sequencer’.
- Alarms from the Detector Controllers and Components Controllers belonging to any instrument-like TCS subsystems to their master controller (for example alarms from the peripheral wavefront sensor controllers to the A&G control system — see [25]).
- Alarms from the OIWFS Components Controller installed in the IOC of an instrument to the A&G GBD Sequencer (see [25] and [10]).

1.1.3.2 Related interfaces governed by ICD/7b, [11]

- Alarms from the TCS subsystems to the TCS Master Control System.

All of the above interfaces are to be implemented using the facilities provided with EPICS and are described in detail below.

While there are details of these interfaces that must differ between their use in the TCS and ICS, the fundamentals of the interfaces should be as consistent as possible to simplify maintenance and flexibility.

1.2 Scope

This document covers only the command, status, and alarm movement within the ICS. The movement of these entities from an ICS to the OCS is covered by the *Gemini Software Design Description* [1], ICD/1a [5], ICD/1b [6] and ICD/2, [7].

The specifics for each individual ICS interface (such as the actual names of the commands and the parameters passed) are not given in this general software ICD. Individual interface details are covered in specific ICDs of the form ICD x.y.z/a.b.c, as described in [4]. ICD 1.9.3/1.9.5, [24], demonstrates how to set out a specific ICD for the Science Instrument to Detector Controller interface.

No bulk data transfer is described here. A description of the bulk data interface between an Instrument Control System and its Detector Controller is given in ICD/3, [8].

No logging events are described here. A description of the interface between an Instrument Control System and its subsystems and the DHS logging system is given in ICD/4, [9].

Only conforming instruments are considered here. The interfaces for non-conforming instruments are described in ICD/8, [12].

The transfer of wavefront information is not included in this ICD, but is described in ICD/5 [10].

The signalling of events on the event bus is not included in this ICD, but is described in ICD/11 [15].

The signalling of interlocks is not included in this ICD, but is described in ICD/12[16].

1.3 Applicable Documents

The following documents should also be consulted:

[1] SPE-C-G0037, *Software Design Description*, Gemini 8m Telescope Project.

[2] SPE-I-G0009, *Software Programming Standards,* Gemini 8m Telescopes Project.

[3] gscg.grp.005, *Gemini System Interfaces*, Gemini 8m Telescopes Project.

[4] *“Instrument (Facilities and Science) ICD’s Guidelines”*, Susan Wieland, Gemini 8m Telescopes Project.

[5] gscg.kkg.009, “*ICD 1a — The System Command Interface”,* Gemini 8m Telescopes Project.

[6] gscg.grp.024, *“ICD 1b — The Baseline Attribute/Value Interface”*, Gemini 8m Telescopes Project.

[7] gscg.kkg.010, “*ICD 2 — Systems Status and Alarm Interfaces”*, Gemini 8m Telescopes Project.

[8] dhs_pdr_icd3, “*ICD 3 — Bulk Data Transfer”*, Norman Hill & Severin Gaudet, DAO.

[9] dhs_pdr_icd4, “*ICD 4 — Logging Interface”*, Norman Hill & Severin Gaudet, DAO.

[10] gscg.grp.010, "*ICD 5 — Wavefront Sensing Information Interface*", Gemini 8m Telescopes Project.

[11] tcs_pbt_004, "*ICD 7b — TCS Subsystem Interfaces*", Philip Taylor, Royal Greenwich Observatory.

[12] gscg.grp.012, "*ICD 8 — Non-Conforming ICS Interfaces*", Gemini 8m Telescopes Project.

[13] gscg.grp.???, "*ICD 9 — EPICS Time Bus Driver*", Phil Taylor, Royal Greenwich Observatory.

[14] gscg.grp.020, "*ICD 10 — EPICS Synchro Bus Driver*", Andrew Johnson, Royal Greenwich Observatory.

[15] gscg.grp.021, "*ICD 11 — Event Bus*", Magnus Paterson, Royal Observatory Edinburgh.

[16] gscg.grp.017, "*ICD 12 — Gemini Interlock System*", Gemini 8m Telescopes Project.

[17] gscg.grp.???, "*ICD 13 — Standard Controller*", Bret Goodrich & Andrew Johnson, Gemini 8m Telescopes Project.

[18] gscg.grp.025, "*ICD 16 — The Parameter Definition Format*", Steve Wampler, Gemini 8m Telescopes Project.

[19] SPE-C-G023, "*The Gemini Standard Controller Hardware Documentation*", Andrew Johnson, Royal Greenwich Observatory.

[20] gscg.sbw.060, "*Implementation Models for the TCS/Subsystem Interfaces*", Steve Wampler, Gemini 8m Telescopes Project.

[21] ocs.kkg.031, "*Sequence Command Specifications*", Kim Gillies, Shane Walker & Steven Beard, NOAO/Gemini 8m Telescopes Project.

[22] cics_smb_002, "*Core Instrument Control System — Introduction & Specification*", Steven Beard, Royal Observatory Edinburgh.

[23] cics_smb_022, "*CICS — Architectural Design Document*", Steven Beard, Royal Observatory Edinburgh.

[24] cics_smb_033, "*ICD 1.9.3/1.9.5 — Science Instrument to Detector Controller*", Steven Beard, Royal Observatory Edinburgh.

[25] ag_jms_001, "*A&G Control System vis-a-vis Other Gemini Systems*", J.M.Stewart, ROE.

[26] ag_jms_004, "*Control of Wavefront Sensors*", J.M.Stewart, ROE.

[27] *WHT ADAM/EPICS INTERFACE*, Version 0.2, Paul Martin, Chris Mayer, Phillip Taylor, 27 July, 1994. Available from the authors.

[28] EPICS *Alarm Handler User's Guide*, Martin Kraimer, Ben-Chin K. Cha, Mark Anderson, Janet Anderson, September, 19, 1991, Los Alamos National Laboratory.

[29] EPICS *Alarm Configuration Tool and Alarm Handler User Interface System Requirements Definition (DRAFT)*, Janet Anderson, Ben Chin-Cha, August 30, 1993. Los Alamos National Laboratory.

[30] EPICS *Display Manager User's Guide*, Los Alamos National Laboratory.

[31] EPICS *Archiving Reference Manual* (DRAFT), Roger Cole, Los Alamos National Laboratory. [N.B. This document is in preparation. The Gemini project has a draft copy. Contact the author for up to date information].

[32] EPICS *Channel Access Reference Manual*, Jeffrey O. Hill, EPICS Group, Los Alamos National Laboratory.

[33] EPICS *Input/Output Controller Record Reference Manual*, J.B. Anderson, M.R. Kraimer, October, 1992.

[34] *EPICS: IOC Application Developers Guide,* Los Alamos National Laboratory.

[35] EPICS *State Notation Language and Run-time Sequencer Users Guide,* Andy Kozubal, Los Alamos National Laboratory

[36] CAPFAST *Electronic Circuit Design CAE User's Manual,* Phase Three Logic Inc., Beaverton, Oregon.

1.4 Abbreviations and Acronyms

A&G	Acquisition and Guidance
AO	Adaptive Optics
CA	Channel Access
CAD	Command Action Directive
CAR	Command Action Response
CC	Components Controller
CICS	Core Instrument Control System
CRCS	Cassegrain Rotator Control System
DC	Detector Controller
DHS	Data Handling System
ECS	Enclosure Control System
EPICS	Experimental Physics and Industrial Control System
FITS	Flexible Image Transport System
GAOS	Gemini Adaptive Optics Subsystem
GBD	Guidance and Beam Direction
GCS	Gemini Control System
ICD	Interface Control Document
ICS	Instrument Control System
IOC	Input Output Controller
IS	Instrument Sequencer
LAN	Local Area Network
M1	Primary Mirror
M2	Secondary Mirror
MCS	Mount Control System
OCS	Observatory Control System
OIWFS	On Instrument Wavefront Sensor
PCS	Primary mirror (M1) Control System
PDF	Parameter Definition Format
SAD	Status/Alarm Database
SCS	Secondary mirror (M2) Control System
SDD	Software Design Description.
SIR	Status/Information Record (EPICS)
TCP/IP	Transmission Control Protocol/Internet Protocol
TCS	Telescope Control System
UDP/IP	User Datagram Protocol/Internet Protocol
VSM	Virtual System Mode
WAN	Wide Area Network
WFS	Wavefront Sensor

1.5 Definitions

VME	A real-time system obeying the ANSI/IEEE 1014-1987 Versatile Backplane Bus standard.
VxWorks	The Real Time Operating system from Wind River

1.6 Stylistic Conventions

References to documents in Section 1.3, “Applicable Documents,” on page 4 are given like this; [1].

Note that, unless otherwise specified, the term “Detector Controller” refers to the Detector Controller *software system*.

2.0 Overview

The Gemini Control System (GCS) is made up of principal software systems: the Telescope Control System (TCS), the Data Handling System (DHS), the Observatory Control System (OCS), and up to four Instrument Control Systems (ICS). In addition, these principal systems contain low-level subsystems which are responsible for a particular area. The TCS is responsible for the following subsystems:

- Acquisition and Guidance subsystem (A&G)
- Gemini Adaptive Optics subsystem (GAOS)
- Cassegrain Rotator Control subsystem (CRCS)
- Enclosure Control subsystem (ECS)
- Mount Control subsystem (MCS)
- Primary mirror (M1) Control subsystem (PCS)
- Secondary mirror (M2) Control subsystem (SCS)

and each ICS is responsible for its own copy of the following subsystems:

- Detector Control subsystem (DC)
- Components Controller subsystem (CC)

An overview of the Gemini Control System is given in the Gemini *Software Design Description*, [1], and an overview of the structure of an Instrument Control System is given in the introduction to the Core Instrument Control System (CICS), [22]. An overview of the A&G and wavefront sensing systems is given in [25] and [26]. This ICD is concerned with the interface between an Instrument Control System and its subsystems, as described in document [3]. The interfaces between the TCS and the TCS subsystems are described in ICD 7b — TCS Subsystem Interfaces, [11].

2.1 System Hardware Architecture

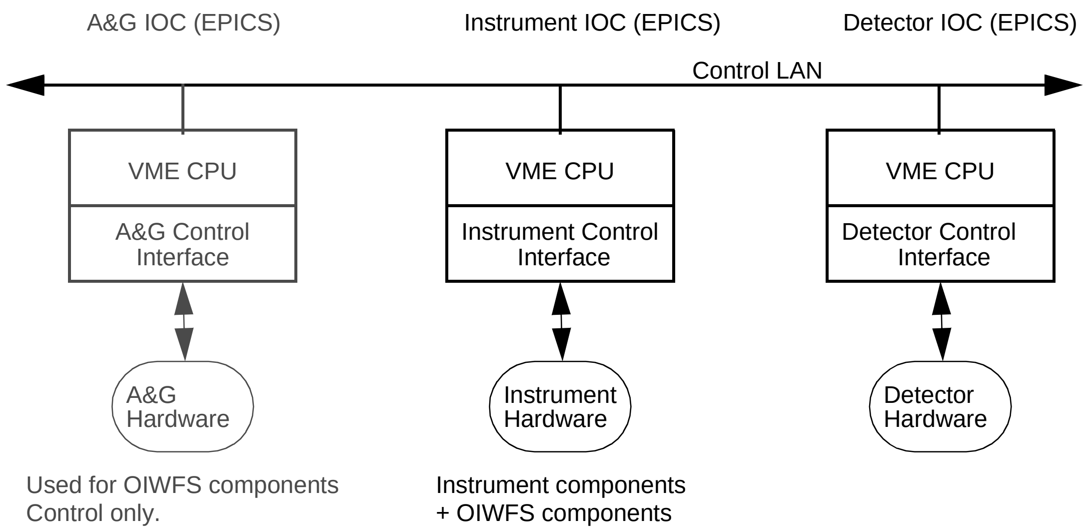
ICD 13, [17], describes the full Gemini system hardware architecture, and gives details on recommended VME hardware. The systems that must conform to this ICD are VxWorks/VME based using EPICS for internal communications and control. Each ICS consists of one or more VxWorks systems supporting the Instrument Sequencer (IS) software, the Detector Controller (DC), and the Components Controller (CC).

In practice, the Instrument Sequencer and Components Controller will all run on the same EPICS Input Output Controller (IOC), and the Detector Controller will be in a separate EPICS IOC, as shown in Figure 1 on page 8. If the Gemini project chooses to use an existing Detector Controller, rather than building one in-house, this might not have the same architecture as shown in Figure 1. It will at least have to act like an EPICS channel access server at the interface with the Control LAN in order to conform to this ICD.

If an instrument has an on-instrument wavefront sensor (OIWFS), the Components Controller for this OIWFS will be installed on the instrument's IOC but will *not* be controlled by the Instrument Sequencer. Instead control will come from the A&G GBD sequencer on a separate IOC, as shown in red on Figure 1. (Note that the Detector Controller for an OIWFS is installed entirely on the A&G IOC). See [25], [26] and [10] for details.

The ICS 'Instrument Sequencer' software is responsible for managing the interfaces between all the instrument systems and other Principal Systems (OCS, DHS and TCS). These interfaces are not covered by this ICD.

FIGURE 1. The Instrument Subsystem Command, Status and Alarm Hardware Architecture



2.2 Communication Architecture

The ICS systems communicate internally using EPICS Channel Access across the Gemini Control LAN. Communications are established using UDP, but run using TCP/IP.

2.2.1 ICS Subsystem Interface Context Diagram

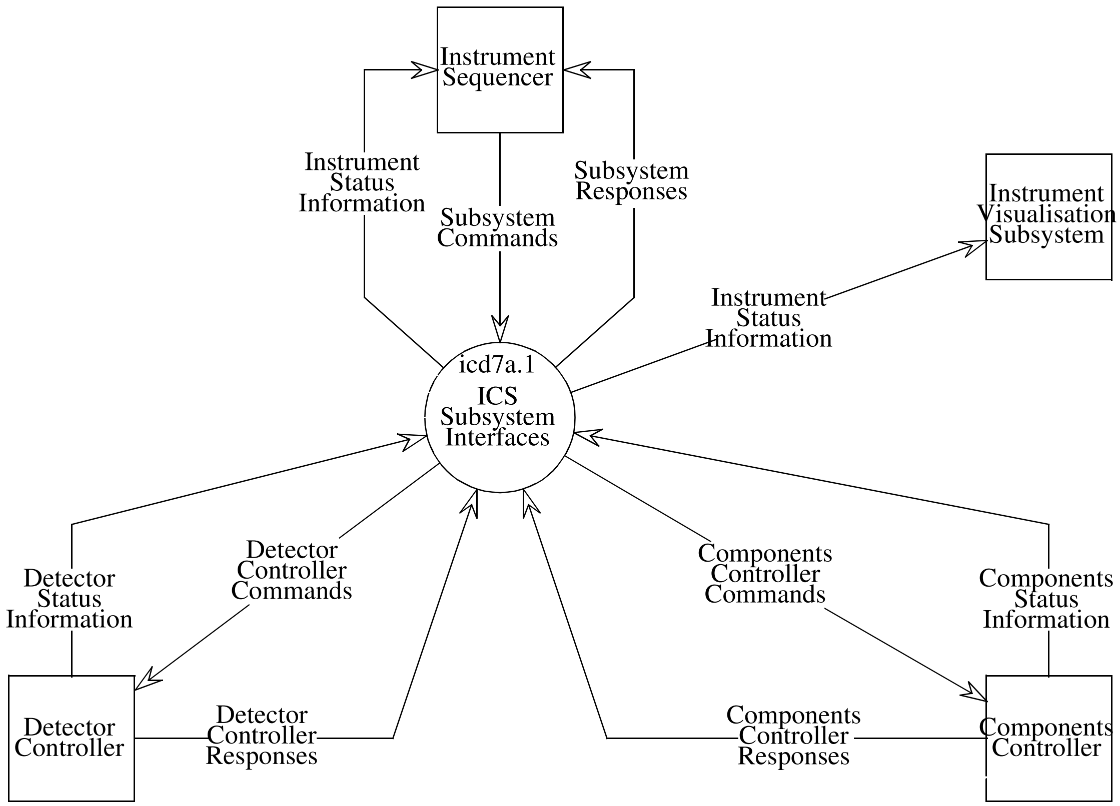
Figure 2 on page 9 shows a context diagram as seen by the ICS subsystem interface. The Gemini systems appear to the interface as entities in the outside world. This diagram may be compared to the "Gemini Control System" data flow diagrams in [1] and [22].

The relationship between this context diagram and the data flow diagrams is as follows:

- All the data flows except those pertaining to the A&G may be found in diagram 2, the “Instrument Control System”.
- The A&G data flows are described in the A&G software documentation, [25] and [26].

FIGURE 2. Context Diagram for a typical ICS Command, Status and Alarm Interface

DfeDiagram - icd7a 9/8/1996 17:21:03



2.2.2 ICS Events and Responses

The events that may happen on the ICS Subsystem Command and Responses interface, and the responses to those events, and shown in Table 1 on page 10

.

TABLE 1. Events and Responses

Event	Response
The ICS Instrument Sequencer issues a command to a subsystem.	The subsystem checks the arguments given with the command and either accepts or rejects it. Acceptance or rejection of the command is acknowledged. (At this point the command initiation transaction is complete). If the command is accepted it is obeyed. Command completion is acknowledged when the command completes (see the event "A subsystem acknowledges command completion." on page 11).
The ICS Instrument Sequencer requests that an active command in a subsystem be paused.	The subsystem checks whether the specified command is active and whether it can be paused, and either accepts or rejects the request. Acceptance or rejection of the request is acknowledged. If the request is accepted the command is paused.
The ICS Instrument Sequencer requests that a paused command in a subsystem be continued.	The subsystem checks whether the specified command is paused and whether it can be continued, and either accepts or rejects the request. Acceptance or rejection of the request is acknowledged. If the request is accepted the command is continued. Command completion is acknowledged in due course (see the event "A subsystem acknowledges command completion." on page 11).
The ICS Instrument Sequencer requests that a command in a subsystem be stopped.	The subsystem checks whether the specified command can be stopped, and either accepts or rejects the request. Acceptance or rejection of the request is acknowledged. If the request is accepted the command is stopped. Command completion is acknowledged with a status indicating the command has been stopped (see the event "A subsystem acknowledges command completion." on page 11).

TABLE 1. Events and Responses

Event	Response
The ICS Instrument Sequencer updates a public process variable contained within the EPICS database belonging to a subsystem.	The subsystem uses the new value in the way it has been programmed to. N.B. The subsystem response depends on how it is set up to use the process variable. In most cases the ICS Instrument Sequencer will already have activated a command (such as MOVE or FOLLOW) requesting that the subsystem monitor the variable, and the variable will contain a parameter (such as a coordinate) which the subsystem must attempt to follow.
The ICS Instrument Sequencer issues a directive to change the simulation level of a subsystem.[a]	The subsystem enters the appropriate simulation level — one of NONE, VSM, FAST, or FULL.
The ICS Instrument Sequencer (or the Instrument Visualisation Subsystem) begins monitoring a process variable in a subsystem.	The subsystem does nothing, as the monitoring is handled transparently by EPICS Channel Access. The ICS Instrument Sequencer receives updates on the value of the process variable. This may be by scan or on changes outside a 'dead-band'.
The ICS Instrument Sequencer (or the Instrument Visualisation Subsystem) begins monitoring the alarm state of a process variable in a subsystem.	The subsystem does nothing, as the monitoring is handled transparently by EPICS Channel Access. The ICS Instrument Sequencer receives any alarms triggered in that process variable.
A subsystem acknowledges command completion.	The ICS Instrument Sequencer receives the acknowledgement, together with an indication of the success or failure of the command and a message describing the command status.
A subsystem changes a process variable.	If a CA client (ICS Instrument Sequencer or Instrument Visualisation Subsystem) has been set up to monitor the state of that process variable then it will receive an update. If no CA clients are monitoring the process variable nothing will be transmitted across the subsystem interface.
A subsystem signals an alarm.	The OCS alarm handler, which should be monitoring the alarm state of all the public process variables in each subsystem, receives the alarm and reports it. If the ICS Instrument Sequencer has been set up to monitor that particular alarm it will also receive notification and can take appropriate action.

TABLE 1. Events and Responses

Event	Response
A subsystem IOC crashes or hangs up.	The OCS alarm handler, which should be monitoring the alarm state of all the public process variables in each subsystem, receives an alarm and reports it. If the ICS Instrument Sequencer is monitoring any process variables or waiting for any outstanding commands it will receive an alarm indicating that the subsystem is no longer connected. If the ICS Instrument Sequencer is not monitoring anything it will get an error the next time it tries to communicate with the subsystem.
A subsystem IOC is rebooted.	The subsystem reloads its EPICS database, then executes its start-up procedure, equivalent to executing the INIT sequence command. It then becomes available again in its start-up state. The ICS Instrument Sequencer can reconnect to a rebooted subsystem without having to be rebooted itself (this is a property of EPICS).

a. Note: This an engineering event. The simulation mode should *not* be changed during a normal observing session. See Section 9.3.1, “Simulation Modes,” on page 21.

Section 3.0, “ICS Subsystem specific Events and Response,” on page 13 gives information on what causes these events, and Section 4.0, “Implementation,” on page 18 gives information on how these various events are implemented. The command interface uses the same protocol as that between the TCS and its subsystems in [11].

Note that only the ICS Instrument Sequencer issues commands. A subsystem can only cause an event by changing a process variable or issuing an alarm. (It can also log information as specified in ICD/4, [9]).

2.3 The Services Component of the Interface

2.3.1 Communication Services and Protocols

All communication takes place using the EPICS Channel Access Protocol [34]. Communication between separate EPICS boxes takes place by means of TCP/IP messages communicated on the Local Area Network.

2.3.2 Host Support Services

These support services are provided through the EPICS Channel Access client software described in [34]

2.3.3 Target Support Services

These support services are provided through the EPICS Channel Access server software described in [34]

3.0 ICS Subsystem specific Events and Response

This section presents the events and responses across this interface for each of the kinds of subsystem. Exact details of the commands and parameters recognised for a particular subsystem can be found in the specific ICD (x.y.z/a.b.c) for that particular subsystem, as described in [4].

3.1 Types of commands

Subsystems need to respond to two kinds of commands — sequence commands and subsystem commands.

3.1.1 Sequence Commands

Sequence commands (`test, reboot, init, datum, apply, verify, endVerify, guide, endGuide, observe, endObserve, pause, continue, stop, abort` and `park`) are commands that all Gemini principal systems must be capable of obeying (even if a system actually does nothing in response to a command). It is also compulsory for a Detector Controller to obey all these commands. It may be convenient for a Components Controller to obey these commands as well, but they are not all compulsory. The commands are described in reference [21]).

3.1.2 Subsystem Commands

Subsystem commands are those which are specific to a given subsystem. They come in two varieties; “One-off” and “Continuous”, as described in [20]:

One-off commands. These are commands which cause some discrete and well-defined action to take place. The action starts, is obeyed and then completes. Examples of one-off commands are requesting that a Components Controller select a particular filter or requesting that a telescope subsystem reset its encoder zero point to a particular value. In general these commands are implemented by issuing a command start request then waiting for a command completion response.

Continuous commands. Continuous commands are those that require a continuous response from the subsystem until further notice. Examples of continuous commands are requesting that a Components Controller should maintain an Atmospheric Dispersion Compensator at a particular position angle on the sky, or that a telescope subsystem should follow some tracking target. In general these commands are executed by sending a continuous stream of demands to a subsystem which is monitoring those demands. The monitoring activity of the subsystem is switched on and off by issuing a “one-off” command.

3.2 Detector Controller (DC) Events and Responses

3.2.1 Sequence commands

A Detector Controller should interpret sequence commands as follows:

TABLE 2. Sequence commands obeyed by a Detector Controller

Sequence Command	Response
test	The DC tests its components (without moving any mechanisms) and reports any failure.
init	The DC initialises itself, obtains the required simulation mode, and downloads its setup information from a specified file. It might also download a set of waveforms. No mechanisms are moved.
datum	If the DC has any moveable components with incremental encoders it defines their position by searching for a datum[a] reference; otherwise it ignores this command.
apply	This command is a signal to act on a set of parameters which the OCS has previously set (by writing parameters to EPICS CAD records or by marking them). The DC checks that the defined set of parameters is acceptable and, if acceptable, it changes its configuration (e.g. by adjusting voltages or downloading a new set of waveforms) to match those parameters. If the parameters are not acceptable the command is rejected. Unlike the other sequence commands, which are implemented using EPICS CAD records, this one is implemented using an APPLY record.
verify	This command is a warning that an observer or engineer wishes to adjust some DC parameters manually. The DC gets itself into a state where the manual adjustment of parameters is possible (if necessary) and ensures all its Status and Information Records are up to date. It may unlock some “interactive only” commands as a result of receiving a “verify”, for example.
endVerify	This command is a signal that verification has finished. It is likely that no DC action will be necessary.
guide	The DC carries out any operations that need to be carried out when the telescope begins guiding. It is likely that no action will be necessary.
endGuide	The DC carries out any operations that need to be carried out when the telescope stops guiding. It is likely that no action will be necessary.
observe	The DC begins an observation using the currently-defined observation parameters. (It may do this by opening a shutter or resetting the detector and allowing it to integrate, for example). When the observation has completed the detector is read out and any necessary processing applied to the data. The data are transmitted to a Data Handling System for storage to disk.
endObserve	The DC ignores this command.

TABLE 2. Sequence commands obeyed by a Detector Controller

Sequence Command	Response
pause	The DC pauses an observation if this is possible. It pauses any exposure timer and suspends any data gathering operations. If it is not possible to pause an observation the DC may reject this command.
continue	The DC resumes an observation which has been paused.
stop	The DC terminates an observation as soon as it can without invalidating the data. Then it reads and processes the data in exactly the same way as if the observation had completed normally. If it is not possible to terminate an observation prematurely the DC may reject this command.
abort	The DC aborts an observation and throws away the data.
park	The DC resets its components and adopts a state where it can safely be switched off.

a. Also known as “home” or “index”.

TABLE 3. Other compulsory commands obeyed by a Detector Controller

Compulsory Command	Response
reboot	The DC IOC should reboot and carry out the same initialisation actions it would normally do when first switched on.
debug	The DC switches to the requested debugging level (NONE, MIN or FULL).
setDhsInfo	This command defines the DHS quick look stream to be used for subsequent observations.
setWcs	The DC accepts a new set of World Coordinate System parameters from the Instrument Sequencer.

Note that the “simulate” command is no longer compulsory, since the simulation mode is now defined on initialisation.

3.2.2 Subsystem commands

In addition to the sequence commands a Detector Controller may also have some detector-specific commands of its own, such as:

- WFCLEAR; clear waveform store.
- WFLOAD; download a waveform from a file.
- *etc.…*

These commands will be specific to a particular Detector Controller and should be documented in an ICD for that Detector Controller (e.g. see ICD 1.9.3/1.9.5, [24]). The choice of additional commands and the requirements placed on a particular Detector Controller are a decision to be made between the instrument and detector groups.

Unlike telescope subsystems (see [20]), a Detector Controller does not have to obey "continuous" commands. Even complex "shift-and-add" or drift scan operations can be defined using discrete parameters supplied with a "one-off" OBSERVE command.

3.2.3 Other Detector Controller Events

A Detector Controller may receive other events, such as:

- Chopper in position (ICD/11, [15]).
- Interlock demand (ICD/12, [16]).

These events are not covered by this ICD but by the ICDs indicated.

3.3 Instrument Visualization Subsystem Events and Responses

The Instrument Visualisation Subsystem (an optional subsystem which may not be needed) is built from PVWave and/or standard EPICS display tools (e.g. the alarm handler, [28], or display manager [30]), and it does not therefore respond to any Gemini sequence commands. It responds to the "monitor process variable" and "monitor alarm" events listed in Table 1 on page 10.

3.4 Components Controller Events and Responses

3.4.1 Sequence commands

A Components Controller should interpret sequence commands as follows:

TABLE 4. Sequence commands obeyed by a Components Controller

Sequence Command	Response
test	The CC tests its components (without moving any mechanisms) and reports any failure.
init	The CC initialises itself, obtains the required simulation mode, and downloads its setup information from a specified file. It does not move any mechanisms.
datum	The CC ensures the positions of all its moveable mechanisms are defined. If any mechanisms have incremental encoders it searches for the datum[a] reference.
apply	This command is a signal to act on a set of parameters which the OCS has previously set (by writing parameters to EPICS CAD records or by marking them). The CC checks that the defined set of parameters is acceptable and, if acceptable, it changes its configuration (e.g. by moving mechanisms) to match those parameters. If the parameters are not acceptable the command is rejected. Unlike the other sequence commands, which are implemented using EPICS CAD records, this one is implemented using an APPLY record.

TABLE 4. Sequence commands obeyed by a Components Controller

Sequence Command	Response
verify (optional)	This command is a warning that an observer or engineer wishes to adjust some CC parameters manually. The CC gets itself into a state where the manual adjustment of parameters is possible (if necessary) and ensures all its Status and Information Records are up to date. It may unlock some "interactive only" commands as a result of receiving a "verify", for example.
endVerify (optional)	This command is a signal that verification has finished. It is likely that no CC action will be necessary.
guide (optional)	The CC carries out any operations that need to be carried out when the telescope begins guiding. This command will probably be irrelevant if the instrument does not have an on-instrument wavefront sensor.
endGuide (optional)	The CC carries out any operations that need to be carried out when the telescope stops guiding. This command will probably be irrelevant if the instrument does not have an atmospheric dispersion compensator or on-instrument wavefront sensor
observe (optional)	The CC begins an observation using the currently-defined observation parameters. Unless the CC has mechanisms that need to move during the observation, it is likely that no action will be necessary.
endObserve	The CC ignores this command.
pause (optional)	The CC pauses an observation. It is likely that no action will be necessary.
continue (optional)	The OC resumes an observation which has been paused. It is likely that no action will be necessary.
stop (optional)	The CC terminates an observation as soon as it can without invalidating the data. It is likely that no action will be necessary.
abort (optional)	The CC aborts an exposure immediately. It is likely that no action will be necessary.
park	The CC parks its components and adopts a state where it can safely be switched off.

a. Also known as "home" or "index".

TABLE 5. Other compulsory commands obeyed by a Components Controller

Compulsory Command	Response
reboot	The CC IOC should reboot and carry out the same initialisation actions it would normally do when first switched on.
debug	The CC switches to the requested debugging level (NONE, MIN or FULL).

Note that the "simulate" command is no longer compulsory, since the simulation mode is now defined on initialisation.

3.4.2 Subsystem commands

In addition to the sequence commands a components Controller may also have some instrument-specific commands of its own, such as:

- FILTER; set filter wheel to specific position.
- SETMODE; set configuration to a particular instrument mode.
- *etc.…*

These commands will be specific to a particular Components Controller and should be documented either in an ICD for that Components Controller (e.g. subsystem 1.9.4 in [4]). The choice of additional commands is the responsibility of each individual instrument group.

A Components Controller may need to obey "continuous" commands like the TCS subsystems (e.g. for scanning a grating) as well as "one-off" commands.

3.4.3 Other Components Controller Events

A Components Controller may receive other events, such as:

- Interlock demand (ICD/12, [16]).

These events are not covered by this ICD but by the ICDs indicated.

4.0 Implementation

The basic interface has already been implemented by the EPICS develop group at Los Alamos National Laboratory in New Mexico. The following EPICS utilities may be used:

- The EPICS Channel Access transport mechanism (CA), [34], which can be used to access database parameters on VxWorks machines running EPICS. This provides all the functionality required for these interfaces.
- The EPICS Database System ([33], [34] and [35]) functions as the target-side termination of these interfaces.
- Existing EPICS tools for database query, monitoring and control (e.g. [28], [29], [30], [31] and [36]).

Besides the basic channel access implementation this interface also relies on the implementation of:

- Command Action Directive (CAD) and Command Action Response (CAR) records, whose implementation is described in ICD/1b, [6].
- APPLY records (if used), whose implementation is also described in ICD/1b, [6].
- Status Information (SIR) records, whose implementation is described in ICD/2, [7].

5.0 The Programmatic Interface

5.1 General Structure

Access through this interface is to be accomplished using the EPICS Channel Access system to set, monitor, and retrieve fields from EPICS database records, as described in reference [32].

5.2 Commands

5.2.1 Sequence commands and "One-off" commands

These commands are implemented using EPICS CAD and CAR records using the mechanism described in ICD/1b, [6]. APPLY records may also be used but, unlike the high level interface between an instrument and the OCS, the use of APPLY records by a subsystem is optional.

5.2.2 "Continuous" Commands

These commands are activated and deactivated using a "One-off" commands implemented with an EPICS CAD/CAR record pair. Once activated, a subsystem executes a continuous command by monitoring a continuous stream of demands transmitted to an EPICS process variable through Channel Access.

5.3 Attributes/Parameters

Attributes and parameters are passed for one-off commands using the argument field of the appropriate CAD record. Parameters for continuous commands are passed as a stream of demands using EPICS process variables.

5.4 Status Information

Status information is exchanged across this interface through public EPICS Status and Information Records (SIR), which are described in ICD/2, [7].

The status information provided by each subsystem should be declared using the Parameter Definition Format (PDF) as specified in ICD/16, [18].

5.5 Data Structures

The data structures transmitted across this interface are the values for EPICS process variables.

6.0 Debugging

6.1 Compiling Programs for Debugging

See [34] for information on compiling and linking programs that use EPICS.

See [2] for information on how to compile and link Gemini software for debugging.

6.2 Debugging Modes

The ICS Instrument Sequencer may direct any subsystem to operate in any of the debugging modes described in ICD 1a, [5].

6.3 Booting and Starting

The procedures for booting and starting a VxWorks system running EPICS are described in the ICD 13, [17], and EPICS IOC developers guide, [34].

7.0 Error, Alarm and Logging System

See ICD 2, [7], for details on reporting errors and raising alarms, and ICD 4, [9], for a description of the history log for recording messages.

8.0 System Attributes

8.1 Maintainability

8.1.1 Interface Design Recommendations

The interface to the ICS subsystems should be through the EPICS database structure and remain consistent with the EPICS database interface presented in [32]. In order to reduce the complexity of maintaining this database, the database should be developed, modified, and maintained using the CAPFAST EPICS database development tool, [36]. The design should reflect the hierarchy that is natural to the application.

To simplify maintenance the interface should obey the following rules:

- All messages should be exchanged by EPICS channel access only. Database access should not be used.
- Status information should be made available through public SIR records.
- Each system's public information should be documented in a Parameter Definition Format (PDF), as specified in ICD/16, [18].

8.1.2 Adaptability and Enhancement Potential

If additional EPICS database records and support code are needed, then these should be discussed with the Gemini Project Office to avoid duplicating work done elsewhere.

9.0 Development and Test Factors

9.1 Project Control

Final authority over the suitability of a particular application for this interface resides with the Gemini Project Office.

9.2 Deliverables

The interfaces described in this report are all directly implemented using EPICS. Consequently, applications of this interface should provide the appropriate EPICS database, a full description of the functionality accessible through the interface, and any support routines that are required.

9.3 Acceptance Testing

Gemini systems must be able to run in a mode that allows their communication with other Gemini systems to be tested. The simulator should mimic the behaviour of this interface.

9.3.1 Simulation Modes

See ICD 1a, [5], for a description of the simulation modes supported by Gemini.