SECS/GEM

Overview

The SECS/GEM (SEMI Equipment Communications Standard/Generic Equipment Model) Module enables Ignition projects and third-party applications to communicate with semiconductor fab equipment. The module is an implementation of the GEM standard (SEMI E30-0307), the SECS-II standard (SEMI E5-0712) and the HSMS standard (SEMI E37-0308).

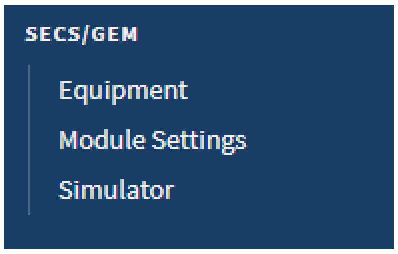
On this page ...

SECS/GEM Host

The module is implemented as a host that can talk to one or more tools via equipment connections configured within the Gateway. While it has some built-in simulation functionality to aid with initial setup, the module serves as a host only, and cannot respond as if were equipment.

Communication with equipment is in the form of messages defined by the SECS-II standard and referred to as SECS messages or just messages. By use of a simple definition file, any SECS message (including custom ones) can be supported.

Once the module is installed, a new header is available under the Config section of the Gateway Webpage that allows you to configure connections to the equipment, or setup simulators that are used for testing.



Equipment Connections

The equipment connections are what allow Ignition to communicate with the specialized semiconductor fab equipment. Equipment connections are setup within the Configure section of the Gateway Webpage. Connections can be made over Ethernet using TCP/IP (HSMS protocol) or over a direct serial connection (SECS-I protocol) using a RS-232 cable. Note, that the direct serial connection requires that the equipment be connected directly to the Ignition Gateway server, and that the Serial Support Gateway Module be installed. The equipment connections also require the use of a database, which will store sent and received messages for audit purposes, and to hold some configuration data. Each equipment connection connects to one tool and one database and transfers SECS messages between the Gateway to the tool. Any number of equipment connections can be configured in the Gateway as long as your license allows it, and different equipment connections can be configured to use the same database connection or different database connections.

SECS Definition Language (SDL) File

The SDL file is where SECS messages are defined. Any SECS message defined in this file can be sent and received. All SECS messages that are sent and received are validated against this file. If a message is not defined or doesn't match a definition then it is not sent, and it is instead inserted into the Errors table as a validation error along with information about why it didn't validate.

Each Equipment Connection has one SDL file and a default SDL file will be used if an alternate is not specified. The SDL file definitions are based on the GEM and SECS standards.

Simulators

The SECS/GEM Module includes the capability to create and configure equipment simulators. The simulators are not a full implementation of the GEM standard. Equipment Connections can connect to simulators and they can exchange various messages. A range of SECS messages are supported by the simulators. This capability exists for getting started with the SECS/GEM Module and for testing applications.

SECS/GEM Message Basics

Once a connection has been properly configured to a tool, messages can be sent between Ignition and the equipment. These messages are often complex, which is why SECS messages are often in the form of a JSON string, which can contain any number of lists and name/value pairs. The messages are typically sent and received using specialized scripting functions within Ignition, such as system.secsgem.sendRequest and system.secsgem.getResponse, that the SECS/GEM module adds. To send a message, a user calls the system.secsgem.sendMessage function in a Python script within an Ignition Gateway or client, and passes the SECS message as JSON text to the function. A transaction id is returned, which can then be passed to the system.secsgem.getResponse function. This function returns the response from the tool as JSON text. However, some SECS messages can be handled differently, such as those that are sent repeatedly from the equipment like status variables. Those messages can be configured to fire Message Handler scripts in the Gateway or Client when the message is received, which can handle the constant flow of messages in a consistent manner. Alternately, the messages can be configured to create Tags with their values, updating the Tags anytime a new message comes in.

Definitions

The SECS/GEM Module uses some unique terms that are not used in the rest of Ignition:

Term	Description
SECS /GEM	The SECS/GEM Module is named after two standards the SECS-II standard (SEMI E5-0712) and the GEM standard (SEMI E30-0307). The SECS-II standard defines messages that can be exchanged between a host and an equipment. SECS stands for Semi Equipment Communications Standard. The GEM standard defines ways for equipment to implement functionality using SECS messages. GEM is short for Generic Model for Communications and Control of Manufacturing Equipment.
HSMS Protocol	A TCP/IP based message transfer protocol for sending SECS messages. The SECS/GEM Module uses HSMS or SECS-I for sending messages.
Equipment	A semiconductor device that sends and receives SECS messages. This is also called a tool.
Equipment Connection	An Equipment Connection enables a database interface to an equipment. An Equipment Connection has a connection to an equipment and a connection to a database and transfers SECS messages between them. Ignition projects or third-party applications can insert SECS messages into a database table to send them to an equipment, and query database tables to retrieve SECS messages from an equipment.
Stream	A category of SECS messages.
Function	A specific SECS message in a stream.
Request Message	Also primary message. A SECS message having an odd numbered function that originates a communication and may require a response message.
Response Message	Also reply message. Also secondary message. A SECS message with an even numbered function that is a response to a request message.
SDL File	Also SECS Definition Language file. A file consisting of definitions of SECS messages and definitions of items used in SECS messages. This file is used to validate SECS messages that are sent and received and to add various functionality having to do with them.
SECS-1	A serial-based message transfer protocol used for sending SECS messages. The SECS-I standard is SEMI E4.
SECS Message	Also just message. A communication message in the format specified by the SECS-II standard that is used to communicate with an equipment.
Transaction	A request message and its response message if required.

TransactionID	Also TxID. Every request message has a unique 32-bit integer identifier that identifies the transaction. Every response message has the same TransactionID as the request message it responds to. In the HSMS and SECS-1 protocols, this is called System Bytes.

In This Section ...

Using the SECS/GEM Module

Getting Started Tutorial

This tutorial will get you up and running with a demo project that uses the SECS/GEM Module.

Download and Installing the the Module

Go to the Downloads page and under the Device Connectivity section, download the SECS/GEM Module. Once the module is downloaded to your computer, install the module. This is done in the normal way that Ignition modules are installed by going to the Gateway Webpage and selecting **Co nfig > System > Modules**. See the Installing or Upgrading a Module page for more information.

Naming Conventions

Before getting started, it's a good idea to establish a naming convention when setting up your SECS/GEM mdoule so you know at a glance what elements are related to each other. This is helpful when you have multiple pieces of equipment, simulators, databases, and equipment connections with keeping these elements organized.

The following feature is new in Ignition version **8.0.5**
Click here to check out the other new features

Configuring the Database Connection

To setup a database connection in Ignition that you want to use with the SECS/GEM Module, go to the **Config > Database > Connections** s ection on the Gateway Webpage. The Database Connection name that is configured for this example is called **SecsGem**.

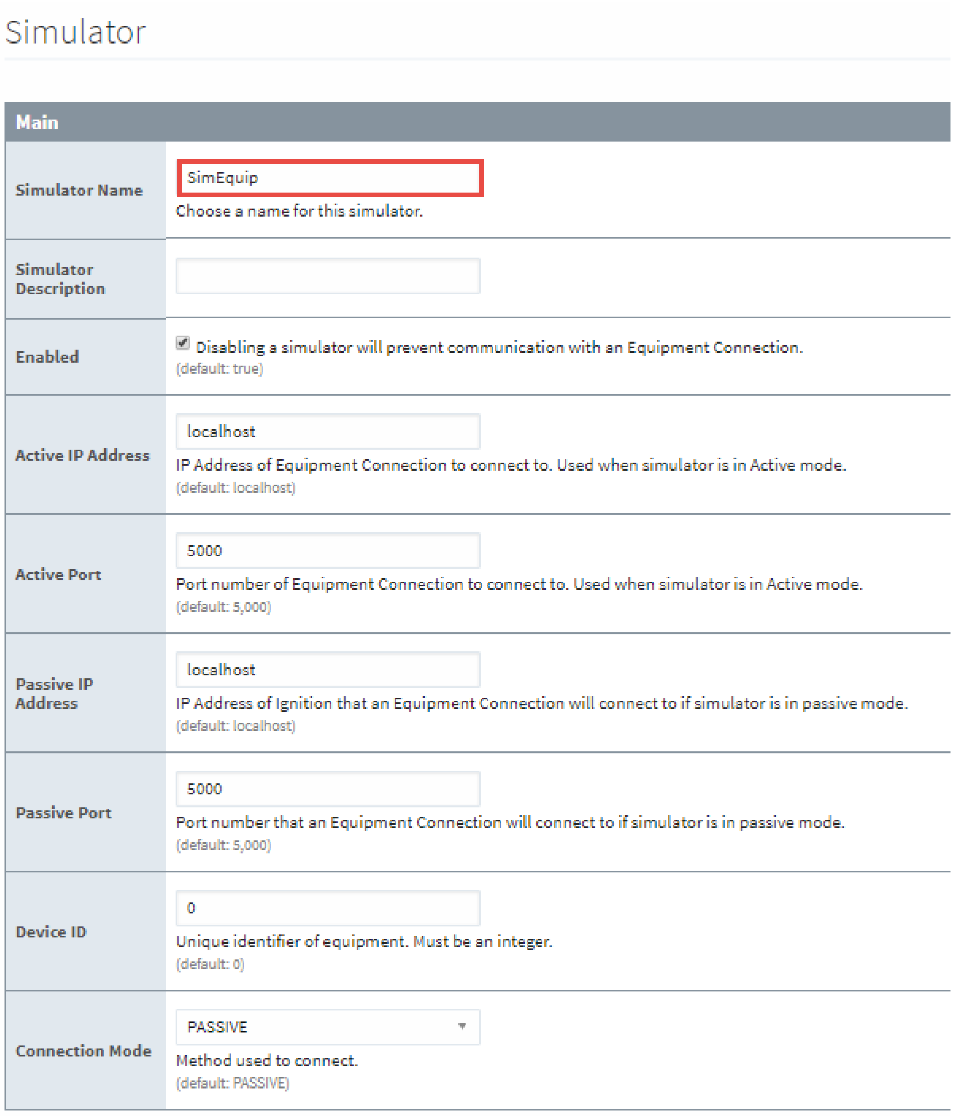
The SECS/GEM Module has been tested with MySQL, SQL and Oracle Servers. The SECS/GEM Demo project that you will upload later as part of this tutorial currently only supports MySQL. More information about configuring a database connection can be found in Connecting to Databases.

Configuring the Simulator

The SECS/GEM Module can directly interact with a piece of equipment, but we can also use the built-in simulator. This tutorial will assume you are using a simulator. To configure a simulator, refer to the example below.

The instructions below will walk you through configuring the simulator. Most of the vaules can be left as defaults to get you up and running quickly.

1. Go to the Gateway Webpage and select **Config > SECS/GEM > Simulator**.
2. On the Simulator page of the Gateway, click on the **Create new Simulator** link.
3. **Simulator Name** - enter the simulator name (i.e., **SimEquip**).
4. **Active Port** - use the default port number of **5000**. The port number needs to be unique if you're using multiple simulators.
5. **Passive Port** - use the default port number of **5000**. The port number needs to be unique if you're using multiple simulators.
6. **Device ID** - use the default ID of **0**. Each piece of equipment must have a unique device ID, and must be an integer.
7. **Connection Mode** - use the default **Passive mode**. The Connection Mode is a property that is configured in both the Simulator and Equipment Connection settings, and cannot be set to the same method. Alternating mode can be used, in which case, it will switch between Active mode and Passive mode until a connection is made.

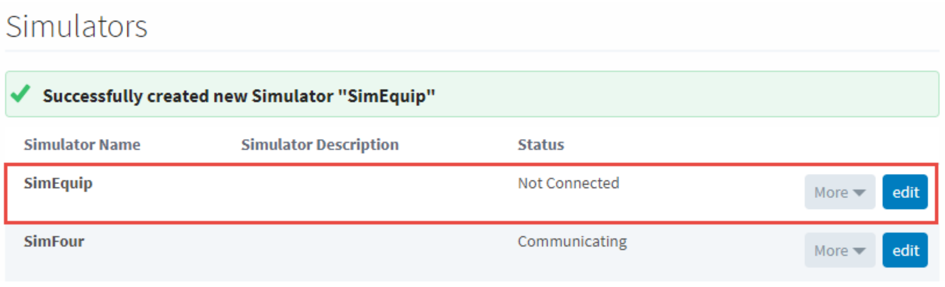


Simulator

Main

Simulator Name	SimEquip Choose a name for this simulator.
Simulator Description	
Enabled	✓ Disabling a simulator will prevent communication with an Equipment Connection. (default: true)
Active IP Address	localhost IP Address of Equipment Connection to connect to. Used when simulator is in Active mode. (default: localhost)
Active Port	5000 Port number of Equipment Connection to connect to. Used when simulator is in Active mode. (default: 5,000)
Passive IP Address	localhost IP Address of Ignition that an Equipment Connection will connect to if simulator is in passive mode. (default: localhost)
Passive Port	5000 Port number that an Equipment Connection will connect to if simulator is in passive mode. (default: 5,000)
Device ID	0 Unique identifier of equipment. Must be an integer. (default: 0)
Connection Mode	PASSIVE Method used to connect. (default: PASSIVE)

8. Click on the **Create New Simulator** button once you entered all your configuration settings. The window will refresh telling you the simulator was created successfully, and the status will toggle back and forth between "Connecting" and "Not Connected" until an equipment connection is created.



Simulators

✓ Successfully created new Simulator "SimEquip"

Simulator Name	Simulator Description	Status	
SimEquip		Not Connected	More edit
SimFour		Communicating	More edit

For more information on configuring a simulator, refer to the SECS/GEM Simulator page.

Trial Mode Reset

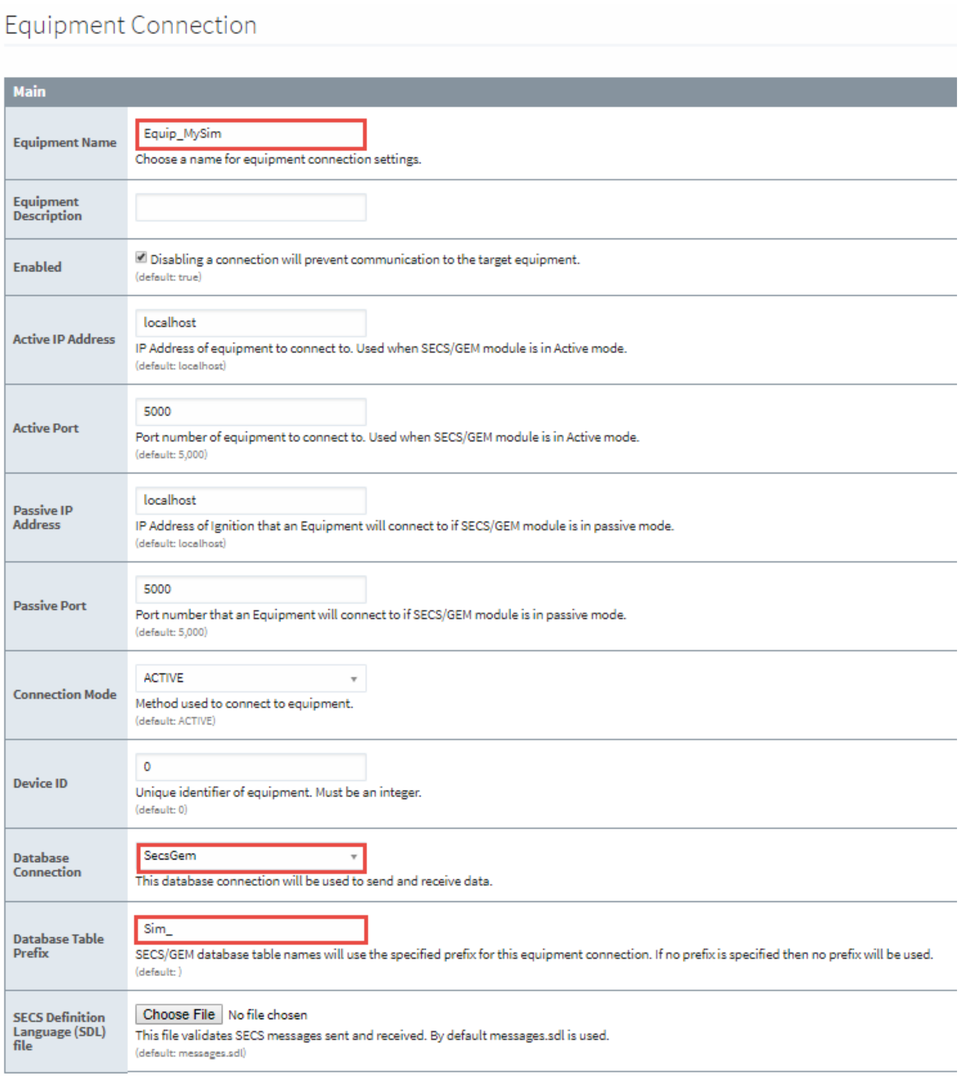
If you do not have a SECS/GEM Module license installed, the simulators and equipment connections will cause communication to be stopped when the Gateway Trial Mode expires in 2 hours. Reset the trial mode on the Gateway Webpage to resume equipment and simulator communications. You can find more information about Trial Mode on the About the Trial Period page.

Creating an Equipment Connection

Once the simulator is configured, (or if there is an actual piece of equipment you wish to connect to), a connection to the Equipment must be made on the **Config > SECS/GEM > Equipment** section of the Gateway Webpage. To configure an equipment connection, refer to the example below.

When creating the equipment connection, most of the settings can be left as the default values for this tutorial. Use the following instructions to configure the equipment connection.

1. On the **Equipment Connections** page, click on **Create new Equipment Connection..**.
2. On the **Add Equipment Connection Step 1: Choose Type** page, select **Ethernet/HSMS Connection**, and click **Next**.
3. On the Equipment Connection page:
    a. **Equipment Name** - enter the equipment name (i.e., **Equip_MySim**).
    b. **Enabled** - default set to **'true'**. Enables communication to the target equipment.
    c. **Active IP Address** - use the default of **localhost.**
    d. **Active Port** - use the default port number of **5000**. The Active Port number must match the port number used in the simulator configuration setup (i.e., 5000).
    e. **Passive IP Address** - use the default of **localhost.**
    f. **Passive Port** - use the default port number of **5000**. The Passive Port number must match the port number used in the simulator configuration setup (i.e., 5000).
    g. **Connection Mode** - use the default **Active mode**.
    h. **Device ID** - the default ID is **0**. Each equipment requires a unique identifier, and must be an integer.
    i. **Database Connection** - enter the database connection name (i.e., **SecsGem)**. This is the database name that was configured in Configuring the Database Connection step.
    j. **Database Table Prefix** - enter a table prefix (i.e., **Sim_**). This is the prefix that will be prepended to each of the database table names that are automatically setup when the connection is made. If no prefix is specified, the table will still get created. The advantage of using a prefix is it helps keep database tables organized, and it is used to differentiate one equipment's tables from another equipment's tables. It also allows you to filter a project based on the prefix.
    k. **SECS Definition Language (SDL) File** - use the default file **messages.sdl**. Validates sent and recieved messages.


Equipment Connection

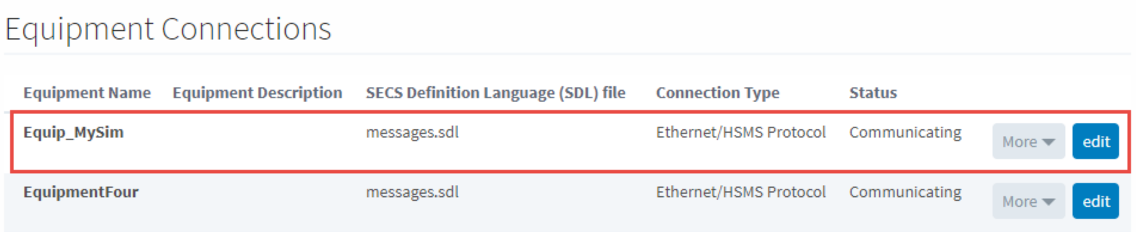
Main	
Equipment Name	Equip_MySim Choose a name for equipment connection settings.
Equipment Description	
Enabled	☑ Disabling a connection will prevent communication to the target equipment. (default: true)
Active IP Address	localhost IP Address of equipment to connect to. Used when SECS/GEM module is in Active mode. (default: localhost)
Active Port	5000 Port number of equipment to connect to. Used when SECS/GEM module is in Active mode. (default: 5,000)
Passive IP Address	localhost IP Address of Ignition that an Equipment will connect to if SECS/GEM module is in passive mode. (default: localhost)
Passive Port	5000 Port number that an Equipment will connect to if SECS/GEM module is in passive mode. (default: 5,000)
Connection Mode	ACTIVE Method used to connect to equipment. (default: ACTIVE)
Device ID	0 Unique identifier of equipment. Must be an integer. (default: 0)
Database Connection	SecsGem This database connection will be used to send and receive data.
Database Table Prefix	Sim_ SECS/GEM database table names will use the specified prefix for this equipment connection. If no prefix is specified then no prefix will be used. (default:)
SECS Definition Language (SDL) file	Choose File No file chosen This file validates SECS messages sent and received. By default messages.sdl is used. (default: messages.sdl)

l. Once all the configuration settings are entered, click on the **Create new Equipment Connection** button. The Equipment Connection will try to connect to the simulator that you setup in Configuring the Simulator section. The window will refresh showing a status of "Connecting" until the equipment connection is created.

More information connecting to a tool can be found on the SECS/GEM Equipment Connections page.

Equipment Connection Status

The status of the Equipment Connection is shown in the Status column on the Equipment Connections webpage. When the status changes to "Communicating", the Equipment Connection successfully connected to the simulator.


Equipment Connections

Equipment Name	Equipment Description	SECS Definition Language (SDL) file	Connection Type	Status	
Equip_MySim		messages.sdl	Ethernet/HSMS Protocol	Communicating	More edit
EquipmentFour		messages.sdl	Ethernet/HSMS Protocol	Communicating	More edit

Download and Restore the Demo Project

At this point, the bare minimum requirements to start utilizing the SECS/GEM module have been met. However, there is a free SECS/GEM Demo Project available from the Ignition Exchange that can be very helpful. This demo project contains resources that allow you to start interacting with the SECS/GEM module.

The demo project can be found here: SECS/GEM Demo Project

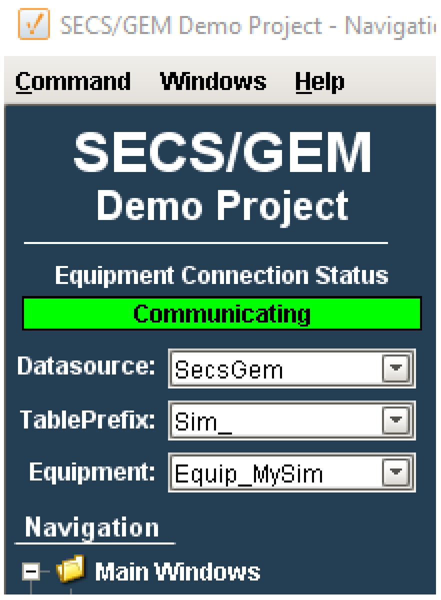
You can learn more about the Ignition Exchange, such as how to import the demo project, on the Ignition Exchange user manual page.

Using the SECS/GEM Demo Project

Once you restored the demo project, launch a client from the Gateway Webpage under the **Home** section to begin using the module. For more information, refer to the page on Launching Clients.

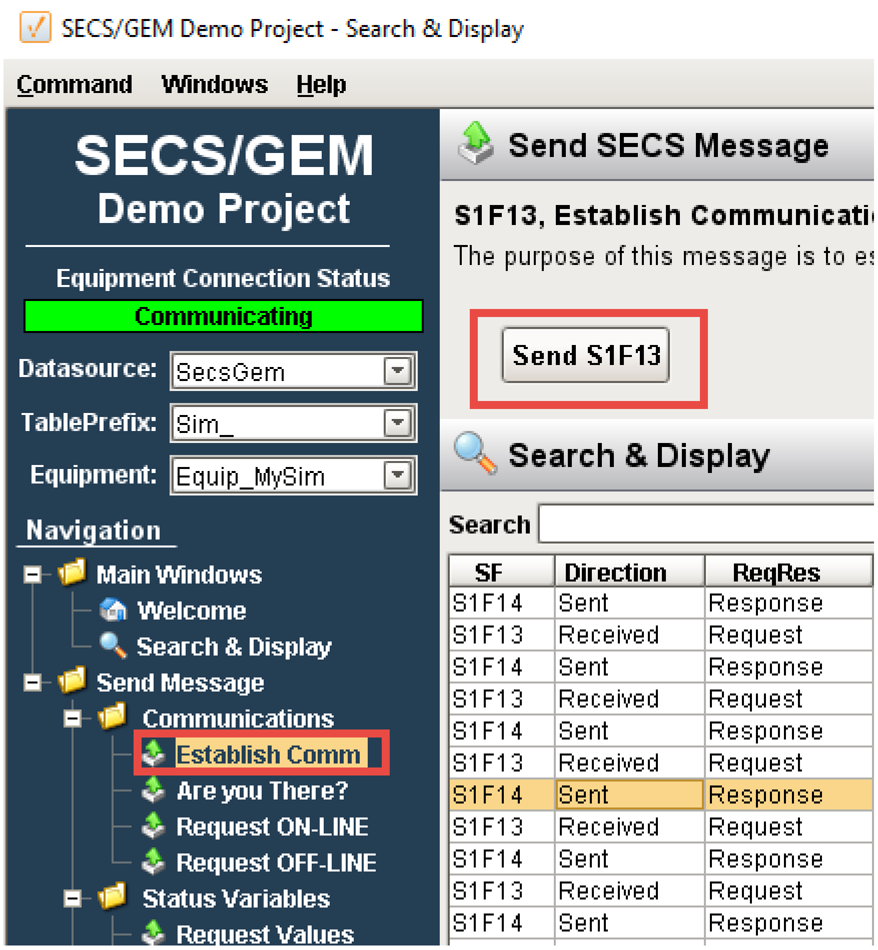
When the client is open, there will be three dropdown menus at the top left of the application. From the dropdowns, make the appropriate selections. If you followed the tutorial, the selections are shown below.

- **Datasource**: select the database connection (i.e., **SecsGem**)
- **TablePrefix**: select the TablePrefix (i.e., **Sim_**).
- **Equipment**: select the name of your Equipment Connection (i.e., **Equip_MySim**).



Datasource: SecsGem
TablePrefix: Sim_
Equipment: Equip_MySim

You should now be able to view and send SECS messages in the application. You can get started by clicking on the **Establish Comm** menu item under the Navigation tree, and then by clicking on the **Send S1F13** button. You will see some messages populate the **Search & Display** table.



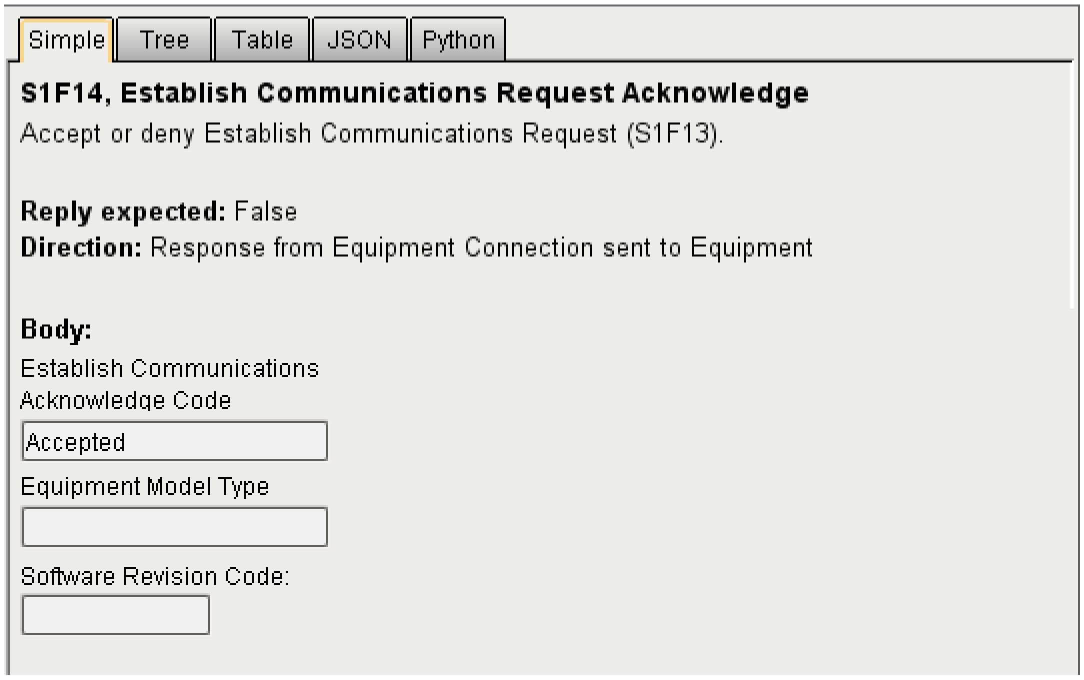
SF	Direction	ReqRes
S1F14	Sent	Response
S1F13	Received	Request
S1F14	Sent	Response
S1F13	Received	Request
S1F14	Sent	Response
S1F13	Received	Request
S1F14	Sent	Response
S1F13	Received	Request
S1F14	Sent	Response
S1F13	Received	Request
S1F14	Sent	Response

Display Tabs

The Display Tabs allow you to see the detailed SECS messages associated with each request and reponse. There are five Display Tabs: Simple, Tree, Table, JSON, and Python. Select any request or response from the Search and Display table and toggle the Display Tabs to view the SECs messages in their respective formats.

The Display Tabs below show the detailed response message (S1F14) returned back from the Establish Comm request (S1F13).

Simple Tab



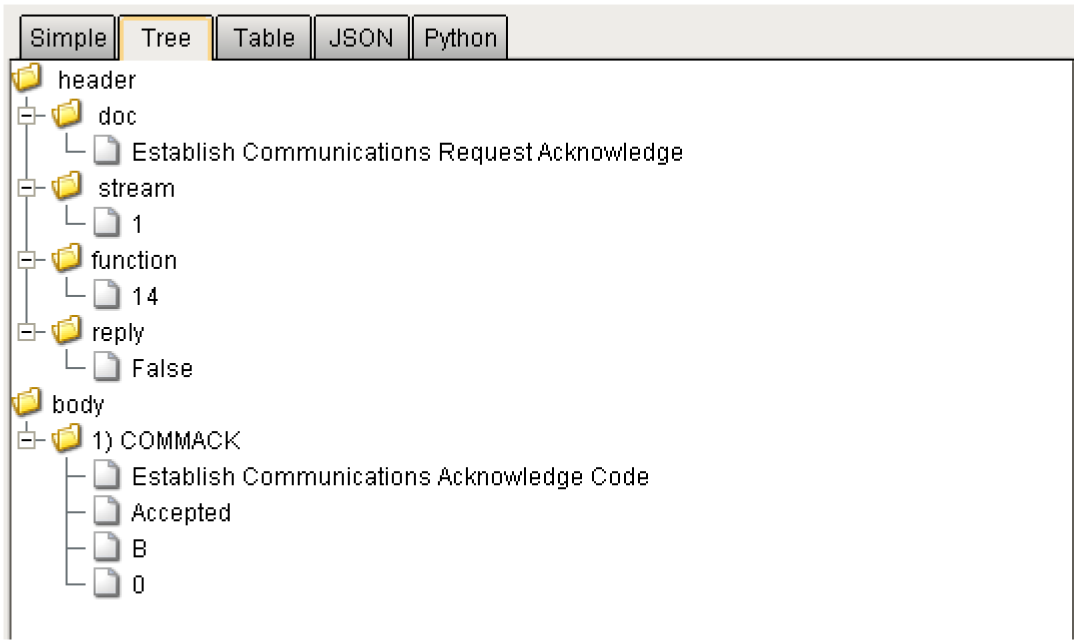
S1F14, Establish Communications Request Acknowledge
Accept or deny Establish Communications Request (S1F13).

Reply expected: False
Direction: Response from Equipment Connection sent to Equipment

Body:
Establish Communications Acknowledge Code
Accepted
Equipment Model Type

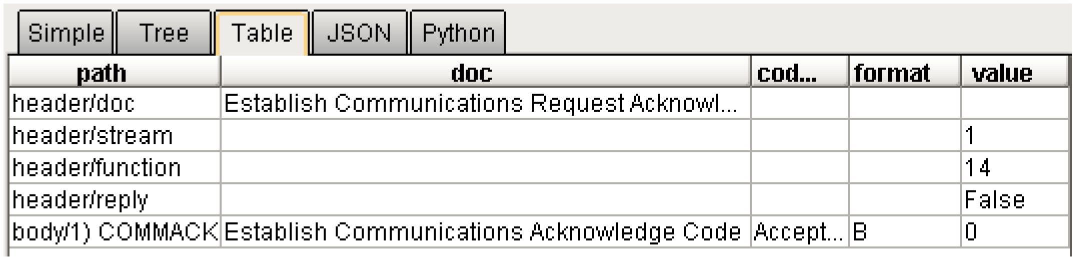
Software Revision Code:

Tree Tab



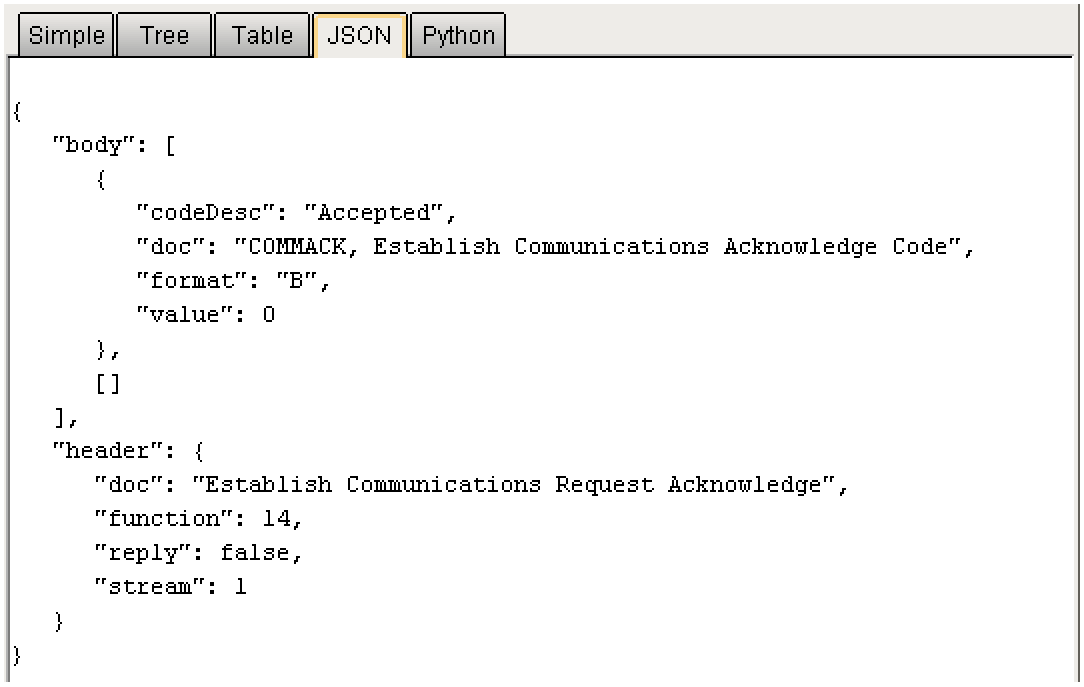
```
header
  doc
    Establish Communications Request Acknowledge
  stream
    1
  function
    14
  reply
    False
body
  1) COMMACK
    Establish Communications Acknowledge Code
    Accepted
    B
    0
```

Table Tab



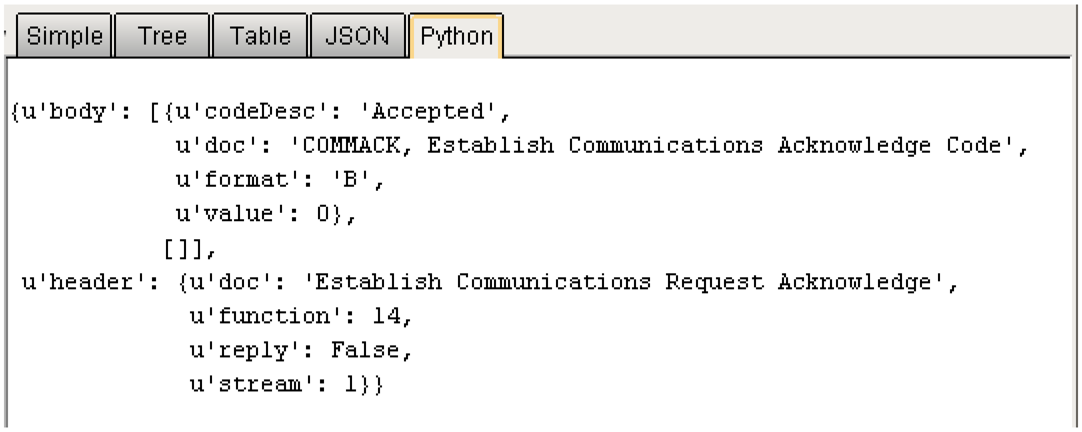
path	doc	cod...	format	value
header/doc	Establish Communications Request Acknowl...			
header/stream				1
header/function				14
header/reply				False
body/1) COMMACK	Establish Communications Acknowledge Code	Accept...	B	0

JSON Tab



```
{
   "body": [
      {
         "codeDesc": "Accepted",
         "doc": "COMMACK, Establish Communications Acknowledge Code",
         "format": "B",
         "value": 0
      },
      []
   ],
   "header": {
      "doc": "Establish Communications Request Acknowledge",
      "function": 14,
      "reply": false,
      "stream": 1
   }
}
```

Python Tab



```
{u'body': [{u'codeDesc': 'Accepted',
            u'doc': 'COMMACK, Establish Communications Acknowledge Code',
            u'format': 'B',
            u'value': 0},
           []],
 u'header': {u'doc': 'Establish Communications Request Acknowledge',
             u'function': 14,
             u'reply': False,
             u'stream': 1}}
```

Related Topics ...

- SECS/GEM Simulator
- SECS/GEM Equipment Connections
- SECS Definition Language (SDL} File

SECS/GEM Equipment Connections

Connection Types

Equipement connections can be created using one of two different connection types: Ethernet through a TCP/IP network, or Serial using an RS-232 serial cable. Each uses a different protocol to communicate.

Ethernet Connections

Ethernet connections use the HSMS protocol for communication. The name of the standard for HSMS is SEMI E37, and are much faster than serial connections. The SECS/GEM module enables ethernet connections automatically and are the preferred method of connection.

Serial Connections

Serial connections use the SECS-I protocol for communication. The name of the standard for SECS-I is SEMI E4. To use a serial connection, the Serial Support Gateway Module must be installed in addition to the SECS/GEM module. The equipment must also have a direct serial connection to the Ignition Gateway server.

On this page ...

Equipment Connection Properties

When setting up an equipment connection through either ethernet or serial, there are various properties that can be configured.

Common Properties

Property Name	Description	Default
Equipment Name	The name to give this equipment connection.	
Equipment Description	A description of this equipment connection.	
Enabled	Whether the equipment connection is enabled or disabled. A disabled equipment connection will prevent communication between the Gateway and the equipment.	true
Device ID	A unique identifier of the equipment, which is defined by the equipment. Must be an integer.	0
Database Connection	The Database Connection that will be used to store messages in for audit purposes as well as some configuration data.	
Database Table Prefix	The prefix that will be added to each of the database table names that get automatically setup when the connection is made. Used to help differentiate one equipment's tables from another equipment's tables. If no prefix is specified, then no prefix will be used.	
SECS Definition Language (SDL) File	The file that validates SECS messages sent and received between the Gateway and the equipment. If none is specified, a default messages.sdl file is used. See SECS Definition Language (SDL) File for the default file.	messages.sdl
T3 Reply Timeout	The number of seconds to wait for an expected SECS message reply.	45

Properties for equipment connections using the Ethernet/HSMS connection type, which connects to equipment using HSMS over TCP.

Property Name	Description	Default
Active IP Address	IP Address of the equipment to connect to when in Active mode.	localhost
Active Port	The Port number of the equipment to connect to when in Active mode, which must match the port number used to create the Simulator.	5000
Passive IP Address	The IP Address of Ignition that an Equipment will connect to if the SECS/GEM module is in passive mode.	localhost
Passive Port	The Port number that an Equipment will connect to if the SECS/GEM module is in passive mode.	5000
Connection Mode	The Method used to connect to Equipment. • ACTIVE- Will attempt to connect to the equipment at the given Active IP Address and Active Port. • PASSIVE - Will listen for a connection from the equipment at the given Passive IP Address and Passive Port. • ALTERNATING - Will switch between Active mode and Passive mode until a connection is made.	ACTIVE
T5 Connect Separation Timeout	The number of seconds which must elapse between successive attempts to connect to the equipment after disconnection.	10
T6 Control Transaction Timeout	The number of seconds which a control transaction (such as LinkTest or Select) may remain open before it is considered a communications failure.	5
T7 Not Selected Timeout	The number of seconds which a TCP/IP connection can remain in NOT SELECTED state (i.e. no HSMS activity) before it is considered a communications failure. This timeout is only used for Passive Connection Mode.	10
T8 Network Intercharacter Timeout	The maximum number of seconds between successive bytes of a single HSMS message before it is considered a communications failure. This applies to HSMS messages received from the equipment.	5
Keep Alive Timeout	The number of seconds interval for sending LinkTest control messages for testing a connection. Automatically reconnects if the test fails. A value of 0 will disable it.	300

Properties for equipment connections using the Serial/SECS-I connection type, which connects to equipment serially using SECS-I.

Property Name	Description	Default
Serial Port Name	The name of the serial port, e.g. "COM1" or "/dev/ttyS0".	COM1
Data Bit Rate	The bit rate of data that is being sent and received. Possible values are: • BR_9600 • BR_4800 • BR_2400 • BR_1200 • BR_19200 • BR_150	BR_9600
T1 Inter-Character Timeout	The maximum number of seconds allowed for interruptions between characters being sent.	0.5
T2 Protocol Timeout	The maximum number of seconds for a lack of protocol response.	10
T4 Inter-Block Timeout	The maximum number of seconds allowed for interruptions in multi-block messages.	45
Retry Limit	The maximum number of send retries allowed.	3

Database Tables

Regardless of the type of equipment connection, database tables are created for each connection based on the prefix specified in the connection if they do not already exist.

No automatic maintenance (such as pruning or partitioning) is performed on these tables. Some of the tables can grow quite large, so it is up to the user to properly manage each set of tables from each equipment connection.

SECSGEM_EquipmentInfo Table

The SECSGEM_EquipmentInfo table is unique in that only one will get created for all databases that use the same database connection and will not use a prefix in its name. Each row of the SECSGEM_EquipmentInfo table contains information about a specific Equipment Connection.

SECSGEM_EquipmentInfo

Column Name	Datatype	Description
Equipment	String	The name of the equipment
Prefix	String	The prefix that the equipment connection uses for its unique tables.
Status	String	The status of the equipment connection.
SDL File	String	The SDL File that was uploaded for this equipment connection. If no file was uploaded, it will use the default messages.sdl.

Messages Table

The Messages table is where SECS messages are stored after they are sent or received. It will include a prefix in the name if one was specified in the equipment connection. Multiple equipment connections can use the same prefix which will have them write to the same messages table. Because this table records every message between the Gateway and equipment, it can grow quite large.

prefixMessages

Column Name	Datatype	Description
ID	Auto incrementing PrimaryID, Integer	The ID of the row.
Equipment	String	The name of the equipment.
StreamFunction	String	The Stream and Function such as "S1F1".
Direction	String	Whether the message was received or sent.
RequestResponse	String	Whether the message was a request or a response.
CommonID	String	A special identifier that will link together multiple related messages.
TxID	Integer	The transaction ID of the message.
Reply	Integer	Whether the message expects a reply or not.
Message	String	The JSON string message.
TimeSentReceived	DateTime	A datetime value of when the message was sent or received.

Errors Table

Any errors that occur are logged to the Errors table. It will include a prefix in the name if one was specified in the equipment connection. Multiple equipment connections can use the same prefix which will have them write to the same errors table.

prefixErrors

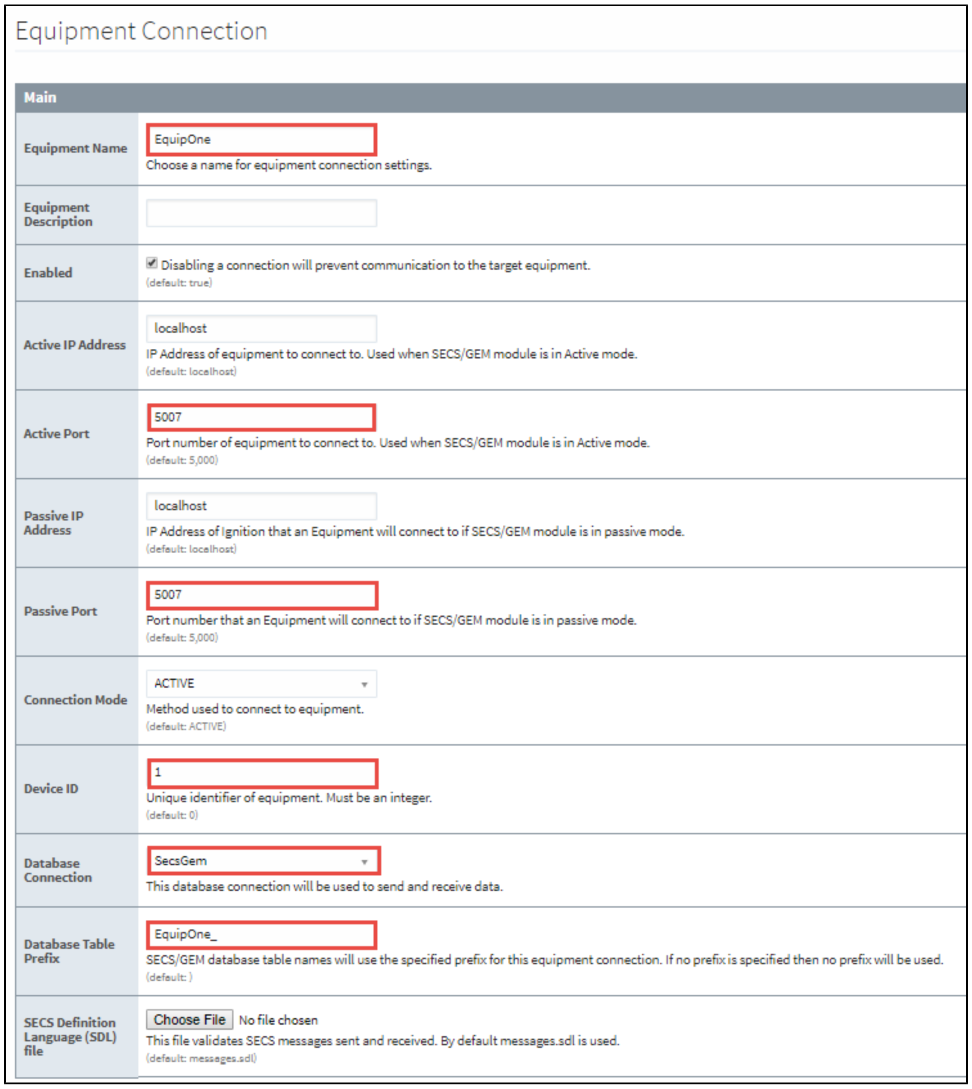
Column Name	Datatype	Description

ID	Auto incrementing PrimaryID, Integer	The ID of the row.
Equipment	String	The name of the equipment.
StreamFunction	String	The Stream and Function such as "S1F1". Only used if the error is related to a SECS Message
ErrorType	String	The type of error that occurred, such as Connection Error, Timeout, or JSON Syntax Error.
Error	String	The error message.
Time	DateTime	The date and time when the error occurred.

Creating an Equipment Connection

In order to communicate with an equipment or the built-in simulator, an Equipment Connection must be configured in the Gateway. This example walks you through configuring an equipment connection for a piece of equipment.

1. Go to the **Configure** section of the Gateway webpage and select **SECS/GEM > Equipment**.
2. On the **Equipment Connections** page, click on **Create new Equipment Connection...**.
3. On the **Add Equipment Connection Step 1: Choose Type** page, select **Ethernet/HSMS Connection**, and click **Next**.
4. On the Equipment Connection page:
    a. **Equipment Name -** enter equipment name, (i.e., **EquipOne**).
    b. **Enabled** - set to **'true'**. Enables communication to the target equipment.
    c. **Active IP Address** - use the default of **localhost**, or enter the IP address of equipment to connect to when in Active mode.
    d. **Active Port** - set the port number (i.e., **5007**). The Active and Passive port numbers must be unique for each piece of equipment. Note, the Active Port must match the port number used in the equipment or simulator configuration setup.
    e. **Passive IP Address** - use the default of **localhost**, or enter the IP address of the equipment to connect to when in Passive mode.
    f. **Passive Port** - set the port number (i.e., **5007)**. The Active and Passive port numbers must be unique for each piece of equipment. Note, the Passive Port must match the port number used in the equipment or simulator configuration setup.
    g. **Connection Mode** - the default is set to **Active**. The Connection Mode is a property that is configured in both the Simulator and Equipment Connection settings, and cannot be set to the same method. Alternating mode can be used, in which case, it will switch between Active and Passive modes until a connection is made.
    h. **Device ID** - enter an equipment ID (i.e.**1**). Each piece of equipment must have a unique ID.
    i. **Database Connection** - enter the database that will be used to send and receive data (i.e, **SecsGem**).
    j. **Database Table Prefix** - enter a prefix to prepend to the database table names. (i.e.,**EquipOne_**). A prefix will be prepended to each of the database table names that are automatically setup when the connection is made. This is an optional setting, so if no prefix is specified, the table will still get created. The advantage of using a prefix is it helps keep database tables organized, and used to differentiate one equipment's tables from another equipment's tables. It also allows you to filter a project based on the prefix.
    k. **SECS Definition Language (SDL) File** - the default file is **messages.sdl**. Validates sent and recieved messages.

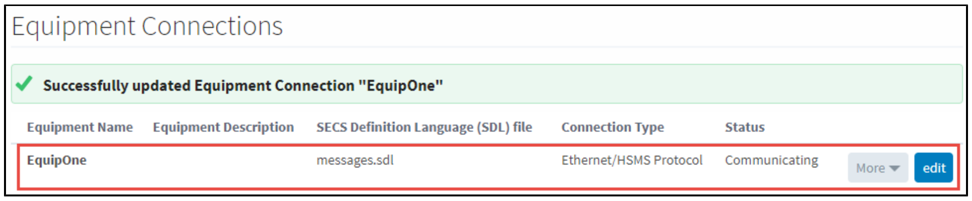


Equipment Connection

Main

Equipment Name	EquipOne Choose a name for equipment connection settings.
Equipment Description	
Enabled	✓ Disabling a connection will prevent communication to the target equipment. (default: true)
Active IP Address	localhost IP Address of equipment to connect to. Used when SECS/GEM module is in Active mode. (default: localhost)
Active Port	5007 Port number of equipment to connect to. Used when SECS/GEM module is in Active mode. (default: 5,000)
Passive IP Address	localhost IP Address of Ignition that an Equipment will connect to if SECS/GEM module is in passive mode. (default: localhost)
Passive Port	5007 Port number that an Equipment will connect to if SECS/GEM module is in passive mode. (default: 5,000)
Connection Mode	ACTIVE Method used to connect to equipment. (default: ACTIVE)
Device ID	1 Unique identifier of equipment. Must be an integer. (default: 0)
Database Connection	SecsGem This database connection will be used to send and receive data.
Database Table Prefix	EquipOne_ SECS/GEM database table names will use the specified prefix for this equipment connection. If no prefix is specified then no prefix will be used. (default:)
SECS Definition Language (SDL) file	Choose File No file chosen This file validates SECS messages sent and received. By default messages.sdl is used. (default: messages.sdl)

5. Once all the configuration settings are entered, click on the **Create new Equipment Connection**. The window will refresh showing a status of "Connecting" until the equipment connection is created. When the status changes to "Communicating", the Equipment Connection is successfully created.



Equipment Connections

✓ **Successfully updated Equipment Connection "EquipOne"**

Equipment Name	Equipment Description	SECS Definition Language (SDL) file	Connection Type	Status	
EquipOne		messages.sdl	Ethernet/HSMS Protocol	Communicating	More edit

Related Topics ...

- SECS Definition Language (SDL) File

SECS Definition Language (SDL) File

The SDL File

The SDL file contains definitions that all SECS messages for that equipment connection are validated against. Any messages that fail to validate against the SDL file will be inserted into the Errors table for the equipment connection instead. The SDL file uses JSON syntax to define datatypes, SECS Items, and SECS Messages.

Each equipment connection can accept one SDL file. However, if no file is specified, the default **'messages.sdl'** file will be used. The SECS message definitions in the default SDL file are based on the definitions in the SECS-II standard (SEMI E5-0712) and the GEM standard (SEMI E30-0307).

The SDL files can be edited in the Gateway Webpage using the built in SDL editor, or they can be downloaded from the Equipment Connections page so that they can be edited in a text editor and re-uploaded at a later time.

On this page ...

SDL File Format

The SDL file should be formatted using JSON syntax. It follows a simple structure, which is demonstrated in the example below. There are four main parts to the SDL file:

doc

The doc comes first and provides any overall documentation for the file. We recommend good documentation so that it is easy to make changes if needed.

formats

The formats list out all of the datatype formats that the items will use. It does not need to include datatypes that are not used by the items.

items

The items are used by the messages to relay information to and from the equipment. The items are sorted alphabetically, with each item under its corresponding letter. Each letter can have any number of items under it.

messages

The messages handle communication with the equipment. Each message is sorted by stream, with each function listed in order under its corresponding stream. There can be multiple streams under messages, and multiple functions under each stream.

Example SDL File

```
{
   "doc":[
      "The doc for the file."
   ],
   "formats":{
      "A":{
         "doc":"ASCII"
      },
      "I2":{
         "doc":"Signed Integer Two Bytes"
      }
   },
   "items":{
      "A":{
         "ALTX":{
            "doc":[
               "ALTX, Alarm Text"
            ],
            "formats":[
               "A"
            ],
            "max":120
         }
      },
      "C":{
         "CCODE":{
            "doc":[
               "CCODE, Command Code",
               "Each command code corresponds to a unique process operation the machine is capable of
performing."
            ],
            "formats":[
               "A",
               "I2",
               "I4",
               "U2",
               "U4"
            ]
         }
      },
      "messages":{
         "S1":{
            "doc":[
               "Stream 1 Equipment Status",
               "This stream provides a means for exchanging information about the status of the",
               "equipment, including its current mode, depletion of various consumable items, and the
status of transfer operations."
            ],
            "S1F0":{
               "doc":[
                  "Abort Transaction",
                  "Used in lieu of an expected reply to abort a transaction.",
                  "Function 0 is defined in every stream and has the same meaning in every stream."
               ],
               "block":"single",
               "direction":"h<->e"
            },
            "S1F1":{
               "doc":[
                  "Are You There Request",
                  "Establishes if the equipment is on-line.",
                  "A function 0 response to this message means the communication is inoperative.",
                  "In the equipment, a function 0 is equivalent to a timeout on the receive timer after
issuing S1,F1 to the host."
               ],
               "block":"single",
               "direction":"h<->e",
               "reply":true,
               "autoReply":{
                  "header":{
                     "stream":1,
                     "function":2,
                     "reply":false
                  },
                  "body":[

                  ]
               }
            }
         }
      }
   }
}
```

SECS Datatype Formats

The formats section should list all datatypes that will be used in the items section. The formats section can be as long as it needs to be, and can even consolidate multiple values into a single value for ease of use. Notice in the example below, the different size integer and floats can all get consolidated into an '`I`' and an '`F` ', which makes it easier to use when specifying all of those datatypes in the Item Definition.

SDL Formats Definition Example

```
{
   "A":{
      "doc":"ASCII"
   },
   "B":{
      "doc":"Binary"
   },
   "L":{
      "doc":"List"
   },
   "I1":{
      "doc":"Signed Integer One Byte"
   },
   "I2":{
      "doc":"Signed Integer Two Bytes"
   },
   "I4":{
      "doc":"Signed Integer Four Bytes"
   },
   "I8":{
      "doc":"Signed Integer Eight Bytes"
   },
   "F4":{
      "doc":"Floating Point Four Bytes"
   },
   "F8":{
      "doc":"Floating Point Eight Bytes"
   },
   "I":{
      "doc":"Signed Integer",
      "formats":[
         "I1",
         "I2",
         "I4",
         "I8"
      ]
   },
   "F":{
      "doc":"Floating Point",
      "formats":[
         "F4",
         "F8"
      ]
   }
}
```

SDL Item Definitions

The SDL Item Definitions list out all possible items that are used in SECS Message definitions. If a SECS Message attempts to use an item that isn't defined in the SDL file, it will throw an error.

SDL Item Definition Example

```
{
   "A":{
      "ACKC7":{
         "doc":[
            "ACKC7, Acknowledge Code"
         ],
         "formats":[
            "B"
         ],
         "bytes":1,
         "codes":{
            "0":"Accepted",
            "1":"Permission not granted",
            "2":"Length error",
            "3":"Matrix overflow",
            "4":"PPID not found",
            "5":"Mode unsupported",
            "6":"Command will be performed with completion signaled later",
            ">":[
               6,
               "Other error"
            ]
         }
      }
   }
}
```

Each item has a list of properties that define what the item is.

Property Name	Description	Example Formatting

doc	Provides information about what the item is along with any other useful information.	`{`
		`   "doc":[`
		`      "TIME, Time of day",`
		`      "12, 16 bytes, or Extended format as specified by the TimeFormat equipment constant value setting.",`
		`      "12-byte format YYMMDDhhmmss",`
		`      "16-byte format YYYYMMDDhhmmsscc",`
		`      "Extended (max 32 byte) format YYYY-MM-DDThh:mm:ss.sTZD (see SEMI E148)"`
		`   ]`
		`}`

formats	A list of the possible datatypes that the item can be.
	
	Can also be a list of items, as in the second example.
	The type can be a list of CATTRDATA objects, or any of the other listed objects.
		`{`
		`   "formats":[`
		`      "A",`
		`      "B",`
		`      "Bool",`
		`      "U",`
		`      "I"`
		`   ]`
		`}`
		
		`{`
		`   "formats":[`
		`      {`
		`         "format":"L",`
		`         "item":"CATTRDATA"`
		`      },`
		`      "B",`
		`      "Bool",`
		`      "A",`
		`      "F",`
		`      "U",`
		`      "I"`
		`   ]`
		`}`

bytes	How many bytes the item must have. If bytes is > 1, then the item is an array of bytes.
	Only applies to single byte datatypes such as Binary, Boolean, U1 and I1.
		`{`
		`        "bytes":1`
		`}`

codes	Lists possible values an item can have and what each means.
	A value of ">" means any value greater than the largest code value.
		`{`
		`   "0":"Accepted",`
		`   "1":"Permission not granted",`
		`   "2":"Length error",`
		`   "3":"Matrix overflow",`
		`   "4":"PPID not found",`
		`   "5":"Mode unsupported",`
		`   "6":"Command will be performed with completion`
		`signaled later",`
		`   ">":[`
		`      6,`
		`      "Other error"`
		`   ]`
		`}`

max	The maximum number of characters that a value can have.
	Only applies to ASCII datatypes.
		`{`
		`   "max":40`
		`}`

pattern	A java regular expression that the ASCII value must match.
	Only applies to ASCII datatypes.
		`{`
		`   "pattern":"^[0-9]{6}|[0-9]{8}$"`
		`}`

SDL Message Definitions

The message definitions list out all possible SECS messages that will be used when communicating with the equipment. There can be many streams listed, each with multiple functions.

SDL Message Definition Example

```
{
   "S6F1":{
      "doc":[
         "Trace Data Send",
         "This function sends samples to the host according to the trace setup done by S2,F23."
      ],
      "block":"multiple",
      "direction":"h<-e",
      "or":[
         {
            "reply":true
         },
         {
            "reply":false
         }
      ],
      "CommonID":[
         "TRID"
      ],
      "realtime":true,
      "SQLTags":true,
      "autoReply":{
         "header":{
            "stream":6,
            "function":2,
            "reply":false
         },
         "body":{
            "format":"B",
            "value":0
         }
      },
      "body":[
         "TRID",
         "SMPLN",
         "STIME",
         [
            "SV",
            "..."
         ]
      ]
   }
}
```

Property Name	Description	Example
doc	Provides information about what the message is along with any other useful information.	`{ "doc":[ "Are You There Request", "Establishes if the equipment is on-line." ] }`
block	The size in bytes of the entire SECS message. There are two possible values: • "single" - The message will not be larger than 254 bytes. • "multiple" - The message will be larger than 254 bytes. This property is only important for maintaining compatibility with the SECS-I protocol through Serial connection.	`{ "block":"single" }`

direction	Determines whether the message should come from the equipment, the host, or either. Possible values are:
	• "h->e"- The message goes from the host to the equipment.
	• "h<-e" - The message goes from the equipment to the host.
	• "h<->e" - The message goes either direction.
		`{`
		`   "direction":"h<->e"`
		`}`

reply	True if the message expects a reply, false if not.	`{`
		`   "reply":true`
		`}`

autoReply	An auto response with a given message when a given message is received.
	
	If included in a message, the example will auto reply with a S1F2 message.
		`{`
		`   "autoReply":{`
		`      "header":{`
		`         "stream":1,`
		`         "function":2,`
		`         "reply":false`
		`      },`
		`      "body":[`
		
		`      ]`
		`   }`
		`}`

body	A list of items that the stream will implement.
	
	The body list can contain lists of items, or the special "...", which signifies a variable list.
	
	In the second example, depending on the value of HCACK, any amount of a list of CNAME and CPACK can be returned.
		`{`
		`   "body":[`
		`      "DATAID",`
		`      "DATALENGTH"`
		`   ]`
		`}`
		
		`{`
		`   "body":[`
		`      "HCACK",`
		`      [`
		`         [`
		`            "CNAME",`
		`            "CPACK"`
		`         ],`
		`         "..."`
		`      ]`
		`   ]`
		`}`

or	Will list multiple possibilities. Can also exist within a body definition.	`{`
		`   "or":[`
		`      {`
		`         "direction":"h->e",`
		`         "body":[`
		
		`         ]`
		`      },`
		`      {`
		`         "direction":"h<-e",`
		`         "body":[`
		`            "MDLN",`
		`            "SOFTREV"`
		`         ]`
		`      }`
		`   ]`
		`}`

Common ID	Used to identify certain messages, especially when used to group multiple incoming messages together. In the example, a trace will have multiple S6F1 messages as the status variable is traced over time. The CommonID will use the trace ID to link all messages from the same trace together, allowing for easy retrieval from the database.	`{ "S6F1":{ "doc":[ "Trace Data Send", "This function sends samples to the host according to the trace setup done by S2,F23." ], "block":"multiple", "direction":"h<-e", "CommonID":[ "TRID" ], "realtime":true, "SQLTags":true, "autoReply":{ "header":{ "stream":6, "function":2, "reply":false } } } }`
realtime	If true, will be processed by a Message Handler script. The message handler must be named onSecsGemRealtimeUpdate. Will use the CommonID for updates. See SECS/GEM Messages for more information on the Message Handler.	`{ "realtime":true }`
SQLTags	If true, will create Tags in the SECSGEM provider automatically, using the CommonID as a folder name. Will create Tags in the location as follows: SECSGEM->EquipmentNameStream and Function->CommonIDmessage value tags Can also be overridden to use a folder name other than the CommonID, like in the second example.	`{ "SQLTags":true }` `{ "SQLTags":[ "SMPLN" ] }`

SDL Message Defaults

The messages section typically starts with a default set of properties before the list of SECS messages. These default property values will be used in the message when the property is not present in the message definition.

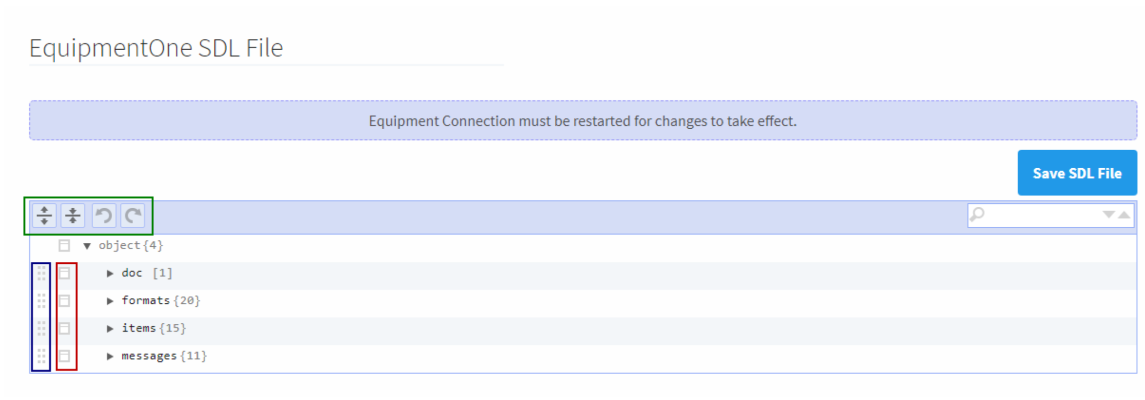
SDL Message Defaults

```
{
   "defaults":{
      "doc":[
         "Default values for properties that are missing."
      ],
      "block":"single",
      "reply":false,
      "direction":"h<->e",
      "body":null
   }
}
```

SDL File Editor

In the **Equipment Connections** section, you can make edits to the equipment's SDL file right in the Gateway Webpage by clicking the **More** button and selecting the **edit sdl file** option.

After editing the SDL file, the equipment connection must be restarted for the changes to take effect.

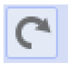


EquipmentOne SDL File

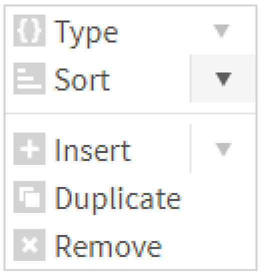
Equipment Connection must be restarted for changes to take effect.

Once in the SDL file editor, there are a few things that can be done. The six dots to the left of each row circled in blue above, allow you to change the order of the rows by clicking and dragging a row and dropping it in a new spot.

The buttons near the top circled in green apply to the document as a whole:

Button	Description
	Expands all nodes.
	Collapses all nodes.
	Undo the last edit made to the file.
	Redo the last Undo.

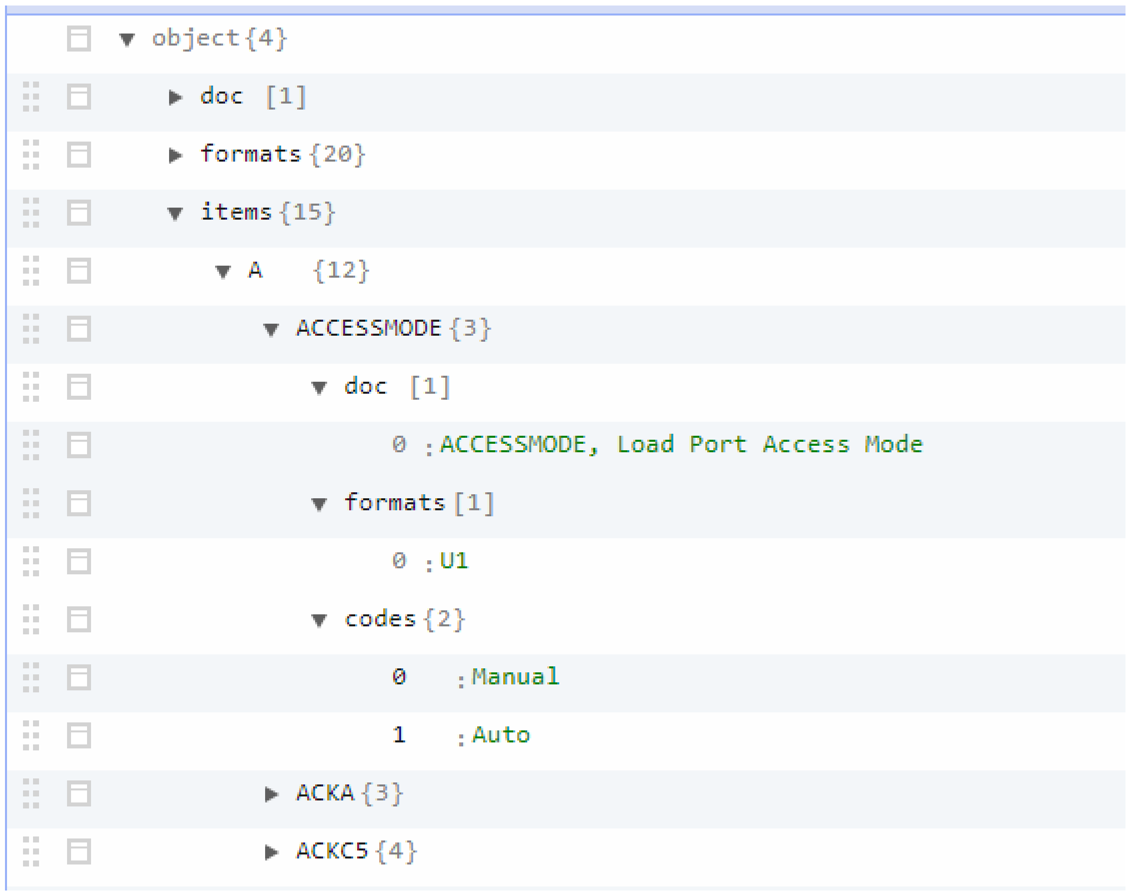
Finally, the buttons circled in red bring up a context menu with a few options that allow you to add and remove rows and change the row type:



Option	Description

Type	Change the type of the row. Possible options are: • Auto - Automatically determines the type based on the value. • Array - Contains an ordered collection of values. • Object - Contains an unordered set of key/value pairs. • String - Determined from the value, but always returned as a string.
Sort	Sorts the rows in either ascending or descending order. • Ascending - Sort the children of this array in ascending order. • Descending - Sort the children of this array in descending order.
Insert	Inserts a new row before the selected row. Possible types are: • Auto - Automatically determines the type based on the value. • Array - Contains an ordered collection of values. • Object - Contains an unordered set of key/value pairs. • String - Determined from the value, but always returned as a string.
Duplicate	Duplicates the row, including all sub elements
Remove	Removes the row and all sub elements.

Each row can be drilled into by clicking on the expand arrow to the left of the element, which allows you to see the sub elements. Many times, these sub elements can also be drilled into to find more sub elements. Once you find the element you want to modify, you can simply edit the text to make a change.



```
▼ object{4}
    ▶ doc  [1]
    ▶ formats{20}
    ▼ items{15}
        ▼ A    {12}
            ▼ ACCESSMODE{3}
                ▼ doc  [1]
                        0 :ACCESSMODE, Load Port Access Mode
                ▼ formats[1]
                        0 :U1
                ▼ codes{2}
                        0    :Manual
                        1    :Auto
            ▶ ACKA{3}
            ▶ ACKC5{4}
```

Once all the edits are made, clicking the **Save SDL File** button at the top right of the screen will save your changes. However, the changes won't take effect until the equipment connection is restarted.

Restart the Equipment

To restart the equipment, simply click the **More** button next to the Equipment on the Equipment Connections page, and select the **'restart'** option.

Default SDL File

The default **'message.sdl'** file is used when no other SDL file is specified.

The messages.sdl Default File.

```
{
    "doc":["SECS stream/functions and items are listed and defined in this document."],

    "formats":{
        "A":{"doc":"ASCII"},
        "B":{"doc":"Binary"},
        "B64":{"doc":"Binary encoded as Base64 string"},
        "J":{"doc":"JIS-8"},
        "L":{"doc":"List"},
        "Bool":{"doc":"Boolean"},
        "Local":{"doc":"Two Byte Character String"},
        "U1":{"doc":"Unsigned Integer One Byte"},
        "U2":{"doc":"Unsigned Integer Two Bytes"},
        "U4":{"doc":"Unsigned Integer Four Bytes"},
        "U8":{"doc":"Unsigned Integer Eight Bytes"},
        "I1":{"doc":"Signed Integer One Byte"},
        "I2":{"doc":"Signed Integer Two Bytes"},
        "I4":{"doc":"Signed Integer Four Bytes"},
        "I8":{"doc":"Signed Integer Eight Bytes"},
        "F4":{"doc":"Floating Point Four Bytes"},
        "F8":{"doc":"Floating Point Eight Bytes"},
        "I":{"doc":"Signed Integer", "formats":["I1","I2","I4","I8"]},
        "U":{"doc":"Unsigned Integer", "formats":["U1","U2","U4","U8"]},
        "F":{"doc":"Floating Point", "formats":["F4","F8"]}},

    "items":{
        "A":{
            "ACCESSMODE":{
                "doc":["ACCESSMODE, Load Port Access Mode"],
                "formats":["U1"],
                "codes":{
                    "0":"Manual",
                    "1":"Auto"}},
            "ACKA":{
                "doc":["ACKA, Indicates success of a request"],
                "formats":["Bool"],
                "codes":{
                    "0":"False",
                    "1":"True"}},
            "ACKC5":{
                "doc":["ACKC5, Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted",
                    ">":[0,"Error, not accepted"]}},
            "ACKC6":{
                "doc":["ACKC6, Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted",
                    ">":[0,"Error, not accepted"]}},
            "ACKC7":{
                "doc":["ACKC7, Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted",
                    "1":"Permission not granted",
                    "2":"Length error",
                    "3":"Matrix overflow",
                    "4":"PPID not found",
                    "5":"Mode unsupported",
                    "6":"Command will be performed with completion signaled later",
                    ">":[6,"Other error"]}},
            "ACKC10":{
                "doc":["ACKC10, Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted for display",
                    "1":"Message will not be displayed",
                    "2":"Terminal not available"}},
            "ALCD":{
                "doc":["ALCD, Alarm Code Byte",
                        "bit 8 = 1 means alarm set",
                        "bit 8 = 0 means alarm cleared",
                        "bit 7-1 is alarm category"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Not Used",
                    "1":"Alarm Cleared, Personal Safety",
                    "2":"Alarm Cleared, Equipment Safety",
                    "3":"Alarm Cleared, Parameter Control Warning",
                    "4":"Alarm Cleared, Parameter Control Error",
                    "5":"Alarm Cleared, Irrecoverable Error",
                    "6":"Alarm Cleared, Equipment Status Warning",
                    "7":"Alarm Cleared, Attention flags",
                    "8":"Alarm Cleared, Data integrity",
                    "><":[8,128,"Alarm Cleared, Other categories"],
                    "128":"Not Used",
                    "129":"Alarm Set, Personal Safety",
                    "130":"Alarm Set, Equipment Safety",
                    "131":"Alarm Set, Parameter Control Warning",
                    "132":"Alarm Set, Parameter Control Error",
                    "133":"Alarm Set, Irrecoverable Error",
                    "134":"Alarm Set, Equipment Status Warning",
                    "135":"Alarm Set, Attention flags",
                    "136":"Alarm Set, Data integrity",
                    ">":[136,"Alarm Cleared, Other categories"]}},
            "ALED":{
                "doc":["ALED, Alarm Enable/Disable Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    ">":[127,"Enable Alarm"],
                    "<":[128,"Disable Alarm"]}},
            "ALID":{
                "doc":["ALID, Alarm Identification"],
                "formats":["U"]},
            "ALTX":{
                "doc":["ALTX, Alarm Text"],
                "formats":["A"],
                "max":120},
            "ATTRDATA":{
                "doc":["ATTRDATA, Contains a specific attribute value for a specific object."],
                "formats":[{"format":"L","item":"ATTRDATA"},"B","Bool","A","F","U","I"]},
            "ATTRID":{
                "doc":["ATTRID,Identifier for an attribute for a specific type of object."],
                "formats":["A","U"]}},
        "C":{
            "CAACK":{
                "doc":["CAACK, Carrier Action Acknowledge Code"],
                "formats":["U1"],
                "codes":{
                    "0":"Acknowledge, command has been performed",
                    "1":"Invalid command",
                    "2":"Can not perform now",
                    "3":"Invalid data or argument",
                    "4":"Acknowledge, request will be performed with completion signaled later by an
event",
                    "5":"Rejected. Invalid state",
                    "6":"Command performed with errors"}},
            "CARRIERACTION":{
                "doc":["CARRIERACTION, Specifies the action requested for a carrier"],
                "formats":["A"]},
            "CARRIERID":{
                "doc":["CARRIERID, The identifier of a carrier"],
                "formats":["A"]},
            "CATTRDATA":{
                "doc":["CATTRDATA, The value of a carrier attribute"],
                "formats":[{"format":"L","item":"CATTRDATA"},"B","Bool","A","F","U","I"]},
            "CATTRID":{
                "doc":["CATTRID, The name of a carrier attribute"],
                "formats":["A"]},
            "CCODE":{
                "doc":["CCODE, Command Code",
                    "Each command code corresponds to a unique process operation the machine is capable
of performing."],
                "formats":["A","I2","I4","U2","U4"]},
            "CEED":{
                "doc":["CEED, Collection event or trace enable/disable code"],
                "formats":["Bool"],
                "bytes":1,
                "codes":{
                    "0":"Disable",
                    "1":"Enable"}},
            "CEID":{
                "doc":["CEID, Collected Event ID"],
                "formats":["U"]},
            "COMMACK":{
                "doc":["COMMACK, Establish Communications Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted",
                    "1":"Denied, Try Again"}},
            "CPACK":{
                "doc":["CPACK, Command Parameter Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "1":"Parameter Name (CPNAME) does not exist",
                    "2":"Illegal Value specified for CPVAL",
                    "3":"Illegal Format specified for CPVAL",
                    ">":[3,"Other equipment specific error"]}},
            "CPNAME":{
                "doc":["CPNAME, Command Parameter Name"],
                "formats":["A"],
                "max":40},
            "CPVAL":{
                "doc":["CPVAL, Command Parameter Value"],
                "formats":["A","B","Bool","U","I"]}},
        "D":{
            "DATAID":{
                "doc":["DATAID, Data ID"],
                "formats":["U"]},
            "DATALENGTH":{
                "doc":["DATALENGTH, Total bytes to be sent"],
                "formats":["U"]},
            "DRACK":{
                "doc":["DRACK, Define Report Acknowledge Code",
                    "If an error condition is detected the entire message is rejected (i.e., partial
changes are not allowed)."],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted",
                    "1":"Denied. Insufficient space",
                    "2":"Denied. Invalid format",
                    "3":"Denied. At least one RPTID already defined",
                    "4":"Denied. At least VID does not exist",
                    ">":[4,"Other errors"]}},
            "DSPER":{
                "doc":["DSPER, Data sample period.",
                    "DSPER has two allowable formats:",
                    "Format 1: hhmmss, 6 bytes",
                    "Format 2: hhmmsscc, 8 bytes",
                    "Where hh is hours, mm is minutes, ss is seconds, and cc is centiseconds.",
                    "Equipment shall either (1) support only Format 1, or (2) support both Format 1 and
Format 2.",
                    "Equipment shall document which formats it accepts.",
                    "Equipment which supports Format 2 need not necessarily support a minimum DSPER of 1
centisecond,",
                    "nor a trace resolution of 1 centisecond, but equipment suppliers shall document its
trace performance limits."],
                "formats":["A"],
                "pattern":"^[0-9]{6}|[0-9]{8}$"}},
        "E":{
            "EAC":{
                "doc":["EAC, Equipment Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Acknowledge",
                    "1":"Denied. At least one constant does not exist",
                    "2":"Denied. Busy",
                    "3":"Denied. At least one constant out of range",
                    ">":[3,"Other equipment specific error"]}},
            "ECDEF":{
                "doc":["ECDEF, Equipment Constant Default value"],
                "formats":[{"format":"L","item":"ECDEF"},"A","B","Bool","J","F","U","
I"]},
            "ECID":{
                "doc":["ECID, Equipment Constant ID"],
                "formats":["U"]},
            "ECMAX":{
                "doc":["ECMAX, Equipment Constant Maximum value"],
                "formats":["A","B","Bool","J","F","U","I"]},
            "ECMIN":{
                "doc":["ECMIN, Equipment Constant Minimum value"],
                "formats":["A","B","Bool","J","F","U","I"]},
            "ECNAME":{
                "doc":["ECNAME, Equipment Constant Name"],
                "formats":["A"]},
            "ECV":{
                "doc":["ECV, Equipment Constant Value"],
                "formats":[{"format":"L","item":"ECV"},"A","B","Bool","J","F","U","I"]},
            "EDID":{
                "doc":["EDID, Expected Data Identification"],
                "formats":["A","B","U","I"]},
            "ERACK":{
                "doc":["ERACK, Enable/Disable Event Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted",
                    "1":"Denied. At least one CEID does not exist",
                    ">":[1,"Other Errors"]}},
            "ERRCODE":{
                "doc":["ERRCODE, Code identifying an error"],
                "formats":["U"],
                "codes":{
                    "0":"No error",
                    "1":"Unknown object in Object Specifier",
                    "2":"Unknown target object type",
                    "3":"Unknown object instance",
                    "4":"Unknown attribute name",
                    "5":"Read-only attribute - access denied",
                    "6":"Unknown object type",
                    "7":"Invalid attribute value",
                    "8":"Syntax error",
                    "9":"Verification error",
                    "10":"Validation error",
                                    "11":"Object identifier in use",
                                    "12":"Parameters improperly specified",
                                    "13":"Insufficient parameters specified",
                                    "14":"Unsupported option requested",
                                    "15":"Busy",
                                    "16":"Not available for processing",
                                    "17":"Command not valid for current state",
                                    "18":"No material altered",
                                    "19":"Material partially processed",
                                    "20":"All material processed",
                                    "21":"Recipe specification related error",
                                    "22":"Failed during processing",
                                    "23":"Failed while not processing",
                                    "24":"Failed due to lack of material",
                                    "25":"Job aborted",
                                    "26":"Job stopped",
                                    "27":"Job cancelled",
                                    "28":"Cannot change selected recipe",
                                    "29":"Unknown event",
                                    "30":"Duplicate report ID",
                                    "31":"Unknown data report",
                                    "32":"Data report not linked",
                                    "33":"Unknown trace report",
                                    "34":"Duplicate trace ID",
                                    "35":"Too many data reports",
                                    "36":"Sample period out of range",
                                    "37":"Group size too large",
                                    "38":"Recovery action currently invalid",
                                    "39":"Busy with another recovery currently unable to perform the
recovery",
                                    "40":"No active recovery action",
                                    "41":"Exception recovery failed",
                                    "42":"Exception recovery aborted",
                                    "43":"Invalid table element",
                                    "44":"Unknown table element",
                                    "45":"Cannot delete predefined",
                                    "46":"Invalid token",
                                    "47":"Invalid parameter",
                                    "48":"Load port does not exist",
                                    "49":"Load port already in use",
                                    "50":"Missing Carrier",
                                    "32768":"Action will be performed at earliest opportunity",
                    "32769":"Action can not be performed now",
                    "32770":"Action failed due to errors",
                    "32771":"Invalid command",
                    "32772":"Client Alr",
                    "32773":"Duplicate ClientID",
                    "32774":"Invalid ClientType",
                    "32775":"IncompatibleVersions",
                    "32776":"Unrecognized ClientID (Client not currently connected)",
                    "32777":"Failed (Completed Unsuccess-fully)",
                    "32778":"Failed (Unsafe) — External intervention required",
                    "32779":"Sensor-Detected Obstacle",
                    "32780":"Material Not Sent",
                    "32781":"Material Not Received",
                    "32782":"Material Lost",
                    "32783":"Hardware Failure",
                    "32784":"Transfer Cancelled"}},
            "ERRTEXT":{
                "doc":["ERRTEXT, Text string describing the error noted in the corresponding ERRCODE"],
                "formats":["A"],
                "max":120}},
        "G":{
            "GRANT":{
                "doc":["GRANT, Grant code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Permission Granted",
                    "1":"Busy, Try Again",
                    "2":"No Space Available",
                    "3":"Duplicate DATAID",
                    ">":[3,"Equipment Specific Error Code"]}},
            "GRANT6":{
                "doc":["GRANT6, Permission to send"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Permission granted",
                    "1":"Busy, try again",
                    "2":"Not interested",
                    ">":[2,"Other err"]}}},
        "H":{
            "HCACK":{
                "doc":["HCACK, Host Command Parameter Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Acknowledge, command has been performed",
                    "1":"Command does not exist",
                    "2":"Cannot perform now",
                    "3":"At least one parameter is invalid",
                    "4":"Acknowledge, command will be performed with completion signaled later by an
event",
                    "5":"Rejected, Already in Desired Condition",
                    "6":"No such object exists"}}},
        "L":{
            "LENGTH":{
                "doc":["LENGTH, Length of the service program or process program in bytes."],
                "formats":["U"]},
            "LRACK":{
                "doc":["LRACK, Link Report Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Accepted",
                    "1":"Denied. Insufficient space",
                    "2":"Denied. Invalid format",
                    "3":"Denied. At least one CEID link already defined",
                    "4":"Denied. At least one CEID does not exist",
                    "5":"Denied. At least one RPTID does not exist",
                    ">":[5,"Other errors"]}}},
        "M":{
            "MDLN":{
                "doc":["MDLN, Equipment Model Type"],
                "formats":["A"],
                "max":20},
            "MEXP":{
                "doc":["MEXP, Message expected in the form SxxFyy where x is stream and y is function."],
                "formats":["A"],
                "max":6},
            "MF":{
                "doc":["MF, Material Format code",
                    "Items with format A will be a unit identifier for one of the special SECS generic
units, as specified in § 12."],
                "formats":["B","A"],
                "bytes":1,
                "codes":{
                    "1":"Quantities in wafers",
                    "2":"Quantities in cassette",
                    "3":"Quantities in die or chips",
                    "4":"Quantities in boats",
                    "5":"Quantities in ingots",
                    "6":"Quantities in leadframes",
                    "7":"Quantities in lots",
                    "8":"Quantities in magazines",
                    "9":"Quantities in packages",
                    "10":"Quantities in plates",
                    "11":"Quantities in tubes",
                    "12":"Quantities in waterframes",
                    "13":"Quantities in carriers",
                    "14":"Quantities in substrates"}},
            "MHEAD":{
                "doc":["MHEAD, SECS message block header associated with message block in error."],
                "formats":["B"]},
            "MID":{
                "doc":["MID, Material ID"],
                "formats":["B","A"],
                "max":80}},
        "O":{
            "OBJACK":{
                "doc":["OBJACK, Acknowledge code"],
                "formats":["U1"],
                "codes":{
                    "0":"Successful completion of requested data",
                    "1":"Error"}},
            "OBJSPEC":{
                "doc":["OBJSPEC, A text string that has an internal format and that is used to point to
```

```
a specific object instance.",
                    "The string is formed out of a sequence of formatted substrings, each specifying an
object’s type and identifier.",
                    "The substring format has the following four fields:",
                    "object type, colon character ':', object identifier, greater-than symbol '>'",
                    "where the colon character ':' is used to terminate an object type and the \"greater
than\" symbol '>' is used to terminate an identifier field.",
                    "The object type field may be omitted where it may be otherwise determined.",
                    "The final '>' is optional."],
                "formats":["A"]},
            "OBJTYPE":{
                "doc":["OBJTYPE, Identifier for a group or class of objects",
                    "All objects of the same type must have the same set of attributes available."],
                "formats":["A","U"]},
            "OFLACK":{
                "doc":["OFLACK, Acknowledge code for OFF-LINE request"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"OFF-LINE Acknowledge"}},
            "ONLACK":{
                "doc":["ONLACK, Acknowledge code for ON-LINE request"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"ON-LINE Accepted",
                    "1":"ON-LINE Not Allowed",
                    "2":"Equipment Already ON-LINE"}}},
        "P":{
            "PARAMNAME":{
                "doc":["PARAMNAME, The name of a parameter in a request"],
                "formats":["A"]},
            "PARAMVAL":{
                "doc":["PARAMVAL, The value of the parameter named in PARAMNAME",
                    "Values that are lists are restricted to lists of single items of the same format
type."],
                "formats":[{"format":"L","item":"PARAMVAL"},"A","B","Bool","F","U","I"]},
            "PORTACTION":{
                "doc":["PORTACTION, tThe action to be performed on a port"],
                "formats":["A"]},
            "PPARM":{
                "doc":["PPARM, Process Parameter"],
                "formats":["A","Bool","F","U","I"]},
            "PPBODY":{
                "doc":["PPBODY, Process Program Body",
                "The process program describes to the equipment, in its own language, the actions to be
taken in processing the material it receives."],
                "formats":["A","B","B64","U","I"]},
            "PPGNT":{
                "doc":["PPGNT, Process Program Grant status"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"OK",
                    "1":"Already have",
                    "2":"No space",
                    "3":"Invalid PPID",
                    "4":"Busy, try later",
                    "5":"Will not accept",
                    ">":[5,"Other error"]}},
            "PPID":{
                "doc":["PPID, Process Program ID",
                    "The format used in the PPID will be host dependent.",
                    "For internal use of the equipment, the PPID can be treated as a unique B pattern.",
                    "If the local equipment is not prepared to display the transmitted code, the display
should be in hexadecimal form."],
                "formats":["A"],
                "max":120},
            "PRJOBID":{
                "doc":["PRJOBID, Text string which uniquely identifies a process job"],
                "formats":["A"]},
            "PRPAUSEEVENT":{
                "doc":["PRPAUSEEVENT, The list of event identifiers, which may be sent as an attribute
value to a process job.",
                    "When a process job encounters one of these events it will pause, until it receives
```

```
the PRJobCommand RESUME."],
                "formats":[{"format":"L","item":"CEID"}]},
            "PRPROCESSSTART":{
                "doc":["PRPROCESSSTART, Indicates that the process resource start processing immediately
when ready."],
                "formats":["Bool"],
                "codes":{
                    "0":"Manual Start",
                    "1":"Automatic Start"}},
            "PRRECIPEMETHOD":{
                "doc":["PRRECIPEMETHOD, Indicates the recipe specification type, whether tuning is
applied and which method is used."],
                "formats":["U1"],
                "codes":{
                    "1":"Recipe only",
                    "2":"Recipe with variable tuning"}},
            "PRSTATE":{
                "doc":["PRSTATE, Enumerated value"],
                "formats":["U1"]},
            "PTN":{
                "doc":["PTN, Material Port number"],
                "formats":["B","U1"],
                "bytes":1}},

        "R":{
            "RCMD":{
                "doc":["RCMD, Remote Command"],
                "formats":["A"]},
            "RCPPARNM":{
                "doc":["RCPPARNM, The name of a recipe variable parameter."],
                "formats":["A"],
                "max":256},
            "RCPPARVAL":{
                "doc":["RCPPARVAL, The initial setting assigned to a recipe variable parameter."],
                "formats":[{"format":"L","item":"RCPPARVAL"},"A","B","Bool","F","U","I"],
                "max":80},
            "RCPSPEC":{
                "doc":["RCPSEC, Recipe specifier. The object specifier of a recipe."],
                "formats":["A"]},
            "REPGSZ":{
                "doc":["REPGSZ, Reporting Group Size"],
                "formats":["U"]},
            "RPTID":{
                "doc":["RPTID, Report ID"],
                "formats":["U"]}},
        "S":{
            "SHEAD":{
                "doc":["SHEAD, Stored header related to the transaction timer."],
                "formats":["B"]},
            "SLOTID":{
                "doc":["SLOTID, Used to reference material by slot (a position that holds material
/substrates) in a carrier.",
                    "This item may be implemented as an array in some messages."],
                "formats":["U1"]},
            "SMPLN":{
                "doc":["SMPLN, Sample Number"],
                "formats":["U","I"]},
            "SOFTREV":{
                "doc":["SOFTREV, Software Revision Code"],
                "formats":["A"],
                "max":20},
            "STIME":{
                "doc":["STIME, Sample Time",
                    "12 or 16 bytes, or Extended format as specified by the TimeFormat equipment
constant value setting.",
                    "12-byte format: YYMMDDhhmmss",
                    "16-byte format: YYYYMMDDhhmmsscc",
                    "Extended (max 32 byte) format: format YYYY-MM-DDThh:mm:ss.sTZD (see SEMI E148)"],
                "formats":["A"],
                "max":32},
            "SV":{
                "doc":["SV, Status Variable"],
                "formats":[{"format":"L","item":"SV"},"B","Bool","A","J","F","U","I"]},
            "SVID":{
                "doc":["SVID, Status Variable ID",
                    "Status variables may include any parameter that can be sampled in time such as
temperature or quantity."],
                "formats":["U"]},
            "SVNAME":{
                "doc":["SVNAME, Status Variable Name"],
                "formats":["A"]}},
        "T":{
            "TEXT":{
                "doc":["TEXT, a single line of characters."],
                "formats":["A","B","Local","U","I"]},
            "TIACK":{
                "doc":["TIACK, Time Acknowledge Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"OK",
                    "1":"Error, not done"}},
            "TIAACK":{
                "doc":["TIAACK, Equipment Acknowledgement Code"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Everything correct",
                    "1":"Too many SVIDs",
                    "2":"No more traces allowed",
                    "3":"Invalid period",
                    "4":"Unknown SVID specified",
                    "5":"Invalid REPGSZ",
                    ">":[63,"Equipment specified error"]}},
            "TID":{
                "doc":["TID, Terminal number"],
                "formats":["B"],
                "bytes":1,
                "codes":{
                    "0":"Single or main terminal",
                    ">":[0,"Additional terminals at the same equipment"]}},
            "TIME":{
                "doc":["TIME, Time of day",
                    "12, 16 bytes, or Extended format as specified by the TimeFormat equipment constant
value setting.",
                    "12-byte format YYMMDDhhmmss",
                    "16-byte format YYYYMMDDhhmmsscc",
                    "Extended (max 32 byte) format YYYY-MM-DDThh:mm:ss.sTZD (see SEMI E148)"],
                "formats":["A"],
                "max":32},
            "TOTSMP":{
                "doc":["TOTSMP, Total Samples"],
                "formats":["U"]},

            "TRID":{
                "doc":["TRID, Trace Request ID."],
                "formats":["U"]}},
        "U":{
            "UNITS":{
                "doc":["UNITS, Units Identifier"],
                "formats":["A","B"]}},
        "V":{
            "V":{
                "doc":["V, Variable data"],
                "formats":[{"format":"L","item":"V"},"A","B","Bool","J","F","U","I"]},
            "VID":{
                "doc":["VID, Variable ID"],
                "formats":["U"]}}},
    "messages":{

        "defaults":{
            "doc":["Default values for properties that are missing."],
            "block":"single",
            "reply":false,
            "direction":"h<->e",
            "body":null},

        "S1":{
            "doc":["Stream 1 Equipment Status",
                "This stream provides a means for exchanging information about the status of the",
                "equipment, including its current mode, depletion of various consumable items, and the
status of transfer operations."],

            "S1F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S1F1":{
                "doc":["Are You There Request",
                    "Establishes if the equipment is on-line.",
                    "A function 0 response to this message means the communication is inoperative.",
                    "In the equipment, a function 0 is equivalent to a timeout on the receive timer
after issuing S1,F1 to the host."],
                "block":"single","direction":"h<->e","reply":true,
                "autoReply":{"header":{"stream":1,"function":2,"reply":false},"body":[]}},

            "S1F2":{
                "doc":["On Line Data",
                    "Data signifying that the equipment is alive.",
                    "Exception: The host sends a zero-length list to the equipment."],
                "block":"single",
                "or":[
                  {"direction":"h<-e","body":["MDLN","SOFTREV"]},
                  {"direction":"h->e","body":[]}]},

            "S1F3":{
                "doc":["Selected Equipment Status Request",
                    "A request to the equipment to report the values of selected status variables.",
                    "Exception: A zero-length list means report all SVIDs."],
                "block":"single","direction":"h->e","reply":true,
                "body":["SVID","..."]},

            "S1F4":{
                "doc":["Selected Equipment Status Data",
                    "The equipment reports the value of each status variable requested in the order
requested.",
                    "The host remembers the names of values requested.",
                    "Exception: A zero-length list item for SV means that SVID does not exist."],
                "block":"multiple","direction":"h<-e",
                "body":["SV","..."]},

            "S1F11":{
                "doc":["Status Variable Namelist Request",
                    "A request to the equipment to report the name and units of selected status
variables.",
                    "Exception: A zero length means report all SVIDs."],
                "block":"single","direction":"h->e","reply":true,
                "body":["SVID","..."]},

            "S1F12":{
                "doc":["Status Variable Namelist Reply",
                    "The equipment reports to the host the name and units of the requested status
variables.",
                    "Exception: Zero-length A items for both SVNAME and UNITS indicates that the SVID
does not exist."],
                 "block":"multiple","direction":"h<-e",
                 "body":[["SVID","SVNAME","UNITS"],"..."]},

            "S1F13":{
                "doc":["Establish Communications Request",
                        "The purpose of this message is to establish communication between an equipment
and host at an application level.",
                    "The purpose of this message is to provide a formal means of initializing
communications at a logical application level both on",
                    "power-up and following a break in communications. It should be the following any
period where host and Equipment SECS",
                    "applications are unable to communicate. An attempt to send an Establish
Communications Request (S1,F13) should be repeated",
                    "at programmable intervals until an Establish Communications Acknowledge (S1,F14) is
received within the transaction timeout",
                    "period with an acknowledgement code accepting the establishment.",
                    "Exception: The host sends a zero-length list to the equipment."],
                "block":"single","reply":true,
                "autoReply":{"header":{"stream":1,"function":14,"reply":false},"body":[{"format":"B","
value":0},[]]},
                "or":[
                    {"direction":"h->e","body":[]},
                    {"direction":"h<-e","body":["MDLN","SOFTREV"]}]},

            "S1F14":{
                "doc":["Establish Communications Request Acknowledge",
                    "Accept or deny Establish Communications Request (S1F13).",
                    "MDLN and SOFTREV are on-line data and are valid only if COMMACK = 0.",
                    "Exception: The host sends a zero-length list for item 2 to the equipment."],
                "block":"single",
                "or":[
                  {"direction":"h<-e","body":["COMMACK",["MDLN","SOFTREV"]]},
                  {"direction":"h->e","body":["COMMACK",[]]}]},

            "S1F15":{
                "doc":["Request OFF-LINE",
                    "The host requests that the equipment transition to the OFF-LINE state."],
                "block":"single","direction":"h->e","reply":true},

            "S1F16":{
                "doc":["OFF-LINE Acknowledge"],
                "block":"single","direction":"h<-e",
                "body":"OFLACK"},

            "S1F17":{
                "doc":["Request ON-LINE",
                    "The host requests that the equipment transition to the ON-LINE state"],
                "block":"single","direction":"h->e","reply":true},

            "S1F18":{
                "doc":["ON-LINE Acknowledge"],
                "block":"single","direction":"h<-e",
                "body":"ONLACK"},

            "S1F23":{
                "doc":["Collection Event Namelist Request",
                    "This function allows the host to retrieve information about what collection event
IDs are available in the equipment",
                    "and which data values (DVVALs) are valid for each collection event.",
                    "Exception: A zero-length list means send information for all CEIDs."],
                "block":"single","direction":"h->e","reply":true,
                "body":["CEID","..."]},

           "S1F24":{
                "doc":["Collection Event Namelist",
                    "The equipment reports to the host the information of the collection events and
associated VIDs of the CEIDs",
                    "requested with the S1F23 message.",
                    "A listed VID can be conditionally or unconditionally associated with the CEID;",
                    "it is the responsibility of the equipment supplier to document whether the
conditional VIDs are reported with S1F24.",
                    "Exception: When both CENAME and the list of associated VIDs are zero-length items,
this indicates that CEID does not exist."],
                "block":"multiple","direction":"h<-e",
                "body":[["CEID","CENAME",["VID","..."]],"..."]}},

        "S2":{
            "doc":["Stream 2 Equipment Control and Diagnostics",
               "Messages which deal with control of the equipment from the host.",
               "This includes all remote operations and equipment self-diagnostics and calibration but
specifically",
               "excludes the control operations which are associated with material transfer (see Stream
4),",
               "loading of executive and boot programs (Stream 8), and all file and operating system
calls",
               "(Streams 10, 13). See also continuations in Stream 17."],

            "S2F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},
            "S2F13":{
                "doc":["Equipment Constant Request",
                    "A request to the equipment to report the values of selected constants.",
                    "Constants such as for calibration, servo gain, alarm limits, data collection mode, 
and other values that are changed infrequently can be obtained using this message.",
                    "Exception: A zero-length list means report all ECVs according to a predefined 
order."],
                    "block":"single","direction":"h->e","reply":true,
                    "body":["ECID","..."]},

            "S2F14":{
                "doc":["Equipment Constant Data",
                    "The equipment reports the value of each constant requested in the order requested.",
                    "Exception: A zero-length list item for ECV means that ECID does not exist. The list 
format for this data item is not allowed, except in this case."],
                "block":"multiple","direction":"h<-e",
                "body":["ECV","..."]},

            "S2F15":{
                "doc":["New Equipment Constant Send",
                    "Change one or more equipment constants."],
                "block":"single","direction":"h->e","reply":true,
                "body":[["ECID","ECV"],"..."]},

            "S2F16":{
                "doc":["New Equipment Constant Acknowledge",
                    "Acknowledges or denies modification of equipment constants.",
                    "If EAC contains a non-zero error code, the equipment should not change any of the 
ECIDs specified in S2F15."],
                "block":"single","direction":"h<-e",
                "body":"EAC"},

            "S2F17":{
                "doc":["Date and Time Request",
                    "Get the date and time from the equipment.",
                    "Useful to check equipment time base or for equipment to synchronize with the host 
time base."],
                "block":"single","direction":"h<->e","reply":true},

            "S2F18":{
                "doc":["Date and Time Data",
                    "Actual time data from equipment.",
                    "Exception: A zero-length item means no time exists"],
                "block":"single","direction":"h<->e",
                "body":"TIME"},

            "S2F23":{
                "doc":["Trace Initialize Send",
                    "Report the current values of selected status variables at a regular interval of 
time.",
                    "Status variables exist at all times. This function provides a way to sample a 
subset of those status variables as a function of time.",
                    "The trace data is returned on S6,F1 and is related to the original request by the 
TRID Multiple trace requests may be made to",
                    "that equipment allowing it. If equipment receives S2F23 with the same TRID as a 
trace function that is currently in progress,",
                    "the equipment should terminate the old trace and then initiate the new trace. A 
trace function currently in progress may be",
                    "terminated by S2,F23 with TRID of that trace and TOTSMP = 0.",
                    "If S2F23 is multi-block, it must be preceded by the S2F39/S2F40 Inquire/Grant 
transaction. Some equipment may support",
                    "only single-Block S6F1, and may refuse a S2F23 message which would cause a multi-
block S6F1.",
                    "Each equipment shall document its trace performance limits. The Host Computer shall 
not send an S2F23 which exceeds the",
                    "equipment's performance limits, or the equipment may operate incorrectly.",
                    "TRID = Trace Request ID, DSPER = Data Sample Period, TOTSMP = Total samples to be 
made.",
                    "REPGSZ = Reporting Group Size, SVID = Status Variable ID"],
                "block":"multiple","direction":"h->e","reply":true,"CommonID":["TRID"],
                "body":["TRID","DSPER","TOTSMP","REPGSZ",["SVID","..."]]},

            "S2F24":{
                "doc":["Trace Initialize Acknowledge"],
                "block":"single","direction":"h<-e",
                "body":{
                   "doc":["Equipment Acknowledgment Code"],
                   "formats":["B"],
                   "bytes":1,
                   "codes":{
                       "0":"Everything correct",
                       "1":"Too many SVIDs",
                       "2":"No more traces allowed",
                       "3":"Invalid period",
                       "4":"Unknown SVID specified",
                       "5":"Invalid REPGSZ",
                       ">":[63,"Equipment specified error"]}}},

            "S2F29":{
                "doc":["Equipment Constant Namelist Request",
                    "A request to the equipment to report information about selected constants.",
                    "This function allows the host to retrieve basic information about what equipment
constants are available in the equipment.",
                    "Exception: A zero-length list means send information for all ECIDs."],
                "block":"single","direction":"h->e","reply":true,
                "body":["ECID","..."]},

            "S2F30":{
                "doc":["Equipment Constant Namelist",
                    "The equipment reports to the host information about certain constants.",
                    "Exception: Zero-length A items for ECNAME, ECMIN, ECMAX, ECDEF, and UNITS indicates
that the ECID does not exist."],
                "block":"multiple","direction":"h<-e",
                "body":[["ECID","ECNAME","ECMIN","ECMAX","ECDEF","UNITS"],"..."]},

            "S2F31":{
                "doc":["Date and Time Set Request",
                    "Set the date and time in the equipment.",
                    "Useful to synchronize the equipment time with the host time base."],
                "block":"single","direction":"h->e","reply":true,
                "body":"TIME"},

            "S2F32":{
                "doc":["Date and Time Set Acknowledge"],
                "block":"single","direction":"h<-e",
                "body":"TIACK"},

            "S2F33":{
                "doc":["Define Report",
                    "The purpose of this message is for the host to define a group of reports for the
equipment.",
                    "The type of report to be transmitted is designated by a BOOLEAN \"Equipment
Constant\".",
                    "An \"Equipment Constant Value\" of \"False\" means that an \"Event Report\" (S6F11)
will be sent, and a value of \"True\" means",
                    "that an \"Annotated Event Report\" (S6F13) will be sent. If S2F33 is Multi-block,
it must be preceded by the S2F39/S2F40 Inquire/Grant transaction.",
                    "Exception 1: A list of zero-length following DATAID deletes all report definitions
and associated links. See S2F35 (Link Event/Report).",
                    "#Exception 2: A list of zero-length following RPTID deletes report type RPTID. All
CEID links to this RPTID are also deleted."],
                "block":"multiple","direction":"h->e","reply":true,
                "body":["DATAID",[["RPTID",["VID","..."]],"..."]]},

            "S2F34":{
                "doc":["Define Report Acknowledge"],
                "block":"single","direction":"h<-e",
                "body":"DRACK"},

            "S2F35":{
                "doc":["Link Event Report",
                    "The purpose of this message is for the host to link n reports to an event (CEID).",
                    "These linked event reports will default to 'disabled' upon linking.",
                    "That is, the occurrence of an event would not cause the report to be sent until
enabled.",
                    "See S2F37 for enabling reports.",
                    "If S2F35 is Multi-block, it must be preceded by the S2F39/S2F40 Inquire/Grant
transaction.",
                    "#Exception: A list of zero length following CEID deletes all report links to that
event."],
                "block":"multiple","direction":"h->e","reply":true,
                "body":["DATAID",[["CEID",["RPTID","..."]],"..."]]},

            "S2F36":{
                "doc":["Link Event Report Acknowledge",
                    "If an error condition is detected the entire message is rejected (i.e., partial
changes are not allowed)."],
                "block":"single","direction":"h<-e",
                "body":"LRACK"},

            "S2F37":{
                "doc":["Enable/Disable Event Report",
                    "The purpose of this message is for the host to enable or disable reporting for a
group of events (CEIDs).",
                    "The reporting for unlisted (CEIDs) is not affected.",
                    "Exception: A list of zero length following <CEED> means all CEIDs."],
                "block":"single","direction":"h->e","reply":true,
                "body":["CEED",["CEID","..."]]},

            "S2F38":{
                "doc":["Enable/Disable Event Report Acknowledge",
                    "If an error condition is detected the entire message is rejected, i.e., partial
changes are not allowed."],
                "block":"single","direction":"h<-e",
                "body":"ERACK"},

            "S2F39":{
                "doc":["Multi-block Inquire",
                    "If a S2,F23, S2,F33, S2,F35, S2,F45, or S2,F49 message is more than one block, this
transaction must precede the message."],
                "block":"single","direction":"h->e","reply":true,
                "body":["DATAID","DATALENGTH"]},

            "S2F40":{
                "doc":["Multi-block Grant",
                    "Grant permission to send multi-block message."],
                "block":"single","direction":"h<-e",
                "body":"GRANT"},

            "S2F41":{
                "doc":["Host Command Send",
                    "The Host requests the Equipment perform the specified remote command with the
associated parameters."],
                "block":"single","direction":"h->e","reply":true,
                "body":["RCMD",[["CPNAME","CPVAL"],"..."]]},

            "S2F42":{
                "doc":["Host Command Acknowledge",
                    "If command is not accepted due to one or more invalid parameters (i.e., HCACK =
3),",
                    "then a list of invalid parameters will be returned containing the parameter name
and reason for being invalid."],
                "block":"single","direction":"h<-e",
                "body":["HCACK",[["CNAME","CPACK"],"..."]]}},

        "S3":{
            "doc":["Stream 3 Materials Status",
                "The functions of the material status stream are used to communicate information and
actions related to material,",
                "including carriers and material-in-process, time-to-completion information, and
extraordinary material occurrences."],

            "S3F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S3F15":{
                "doc":["Materials Multi-Block Inquire",
                    "This message requests permission to send a multi-block message based upon a maximum
length of the total message.",
                    "It must be sent prior to sending any multi-block primary message in Stream 3."],
                "block":"single","direction":"h->e","reply":true,
                "body":["DATAID","DATALENGTH"]},
            "S3F16":{
                "doc":["Materials Multi-Block Grant",
                    "This message grants or denies permission to send a multi-block primary message in
Stream 3."],
                "block":"single","direction":"h<-e",
                "body":"GRANT"},

            "S3F17":{
                "doc":["Carrier Action Request",
                    "This message requests an action to be performed for a specified carrier.",
                    "If multi-block, this message must be preceded by the S3F15/F16 transaction.",
                    "Exception: If variable list has no items, then no carrier attributes are included.",
                    "If CARRIERID is not a zero-length item, then PTN may be omitted (a zero-length
item)",
                    "ATTRID and ATTRDATA may be substituted for CATTRID and CATTRDATA respectively.",
                    "ReticlePodLocationID may be used as one of <CATTRID> when the CARRIERACTION is
PodRelease and the carrier is not at a Load Port."],
                "block":"multiple","direction":"h->e","reply":true,
                "body":["DATAID","CARRIERACTION","CARRIERID","PTN",[["CATTRID","CATTRDATA"],"..."]]},

            "S3F18":{
                "doc":["Carrier Action Acknowledge",
                    "This message acknowledges the carrier action request.",
                    "Exception: If variable list contains no items then no errors exist"],
                "block":"single","direction":"h<-e",
                "body":["CAACK",[["ERRCODE","ERRTEXT"],"..."]]},

            "S3F25":{
                "doc":["Port Action Request",
                    "This message requests an action be performed for a port.",
                    "Exception: If variable list contains no items, then no parameters are provided."],
                "block":"single","direction":"h->e","reply":true,
                "body":["PORTACTION","PTN",[["PARAMNAME","PARAMVAL"],"..."]]},

            "S3F26":{
                "doc":["Port Action Acknowledge",
                    "This message acknowledges the port action request.",
                    "Exception: If variable list contains no item then no errors exist."],
                "block":"single","direction":"h<-e",
                "body":["CAACK",[["ERRCODE","ERRTEXT"],"..."]]},

            "S3F27":{
                "doc":["Change Access",
                    "The Host requests the Equipment to change the Access Mode for the specified Load
Ports.",
                    "ACCESSMODE specifies the desired Access Mode. PTN specifies a desired Load Port
Number.",
                    "If variable list contains no items then the command applies to all Load Ports on
the equipment.",
                    "If any specified port is already in the specified Access Mode, then the Equipment
shall accept the command,",
                    "and toggle all loadports to specified mode.",
                    "If the Equipment is unable to change one or more of specified Port(s) to the
specified Access Mode,",
                    "then the Equipment shall accept the command (with appropriate response
acknowledgement),",
                    "and shall change only the Access Mode of those Port(s) allowed by the equipment,",
                    "supplying the host with an indication that not all ports were successfully
changed."],
                "block":"single","direction":"h->e","reply":true,
                "body":["ACCESSMODE",["PTN","..."]]},

            "S3F28":{
                "doc":["Change Access Acknowledge",
                    "If the command is successful, CAACK = 0, and variable list size = 0.",
                    "If the command was successful for some ports, CAACK = 6, and variable list size >
0."],
                "block":"single","direction":"h<-e",
                "body":["CAACK",[["PTN","ERRCODE","ERRTEXT"],"..."]]}},

        "S5":{
            "doc":["Stream 5 Exception Handling",
                "This stream contains messages regarding binary and analog equipment exceptions.",
                "Exceptions are classified into two categories: errors and alarms."],
            "S5F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S5F1":{
                "doc":["Alarm Report Send",
                    "This message reports a change in or presence of an alarm condition.",
                    "One message will be issued when the alarm is set and one message will be issued
when the alarm is cleared.",
                    "Irrecoverable errors and attention flags may not have a corresponding clear
message."],
                "block":"single","direction":"h<-e","reply":true,
                "autoReply":{"header":{"stream":5,"function":2,"reply":false},"body":{"format":"B","
value":0}},
                "body":["ALCD","ALID","ALTX"]},

            "S5F2":{
                "doc":["Alarm Report Acknowledge",
                    "Acknowledge or error."],
                "block":"single","direction":"h->e",
                "body":"ACKC5"},

            "S5F3":{
                "doc":["Enable/Disable Alarm Send",
                    "This message will change the state of the enable bit in the equipment.",
                    "The enable bit determines if the alarm will be sent to the host.",
                    "Alarms which are not controllable in this way are unaffected by this message.",
                    "Exception: A zero-length item for ALID means all alarms."],
                "block":"single","direction":"h->e","reply":true,
                "body":["ALED","ALID"]},

            "S5F4":{
                "doc":["Enable/Disable Alarm Acknowledge",
                    "Acknowledge or error."],
                "block":"single","direction":"h<-e",
                "body":"ACKC5"},

            "S5F5":{
                "doc":["List Alarms Request",
                    "This message requests the equipment to send binary and analog alarm information to
the host.",
                    "A zero-length item means send all possible alarms regardless of the state of
ALED."],
                "block":"single","direction":"h->e","reply":true,
                "body":"ALID..."},

            "S5F6":{
                "doc":["List Alarm Data",
                    "This message contains the alarm data known to the equipment.",
                    "A zero-length item returned for ALCD or ALTX means that value does not exist."],
                "block":"multiple","direction":"h<-e",
                "body":[["ALCD","ALID","ALTX"],"..."]}},

        "S6":{
           "doc":["Stream 6 Data Collection",
               "This stream is intended to cover the needs of in-process measurements and equipment
monitoring."],

            "S6F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S6F1":{
                "doc":["Trace Data Send",
                    "This function sends samples to the host according to the trace setup done by
S2F23.",
                    "Trace is a time-driven form of equipment status.",
                    "Even if S6F1 is multi-block, it is not preceded by an Inquire/Grant transaction,
because the Host S2F23 is an implicit grant.",
                    "Some equipment may support only single-block S6F1, and may refuse an S2F23 (Trace
Initiate Send) message which would cause a multi-block S6,F1.",
                    "Exception: A zero-length STIME means no value is given and that the time is to be
derived from SMPLN along with knowledge of the request.",
                    "TRID = Trace Request ID, SMPLN = Sample Number, STIME = Sample Time, SV = Status
Variable Value"],
                "block":"multiple","direction":"h<-e","or":[{"reply":true},{"reply":false}],
                "CommonID":["TRID"],"realtime":true,"SQLTags":true,
                "autoReply":{"header":{"stream":6,"function":2,"reply":false},"body":{"format":"B","
value":0}},
                "body":["TRID","SMPLN","STIME",["SV","..."]]},

            "S6F2":{
                "doc":["Trace Data Acknowledge",
                    "Acknowledge or error."],
                "block":"single","direction":"h->e",
                "body":"ACKC6"},

            "S6F5":{
                "doc":["Multi-block Data Send Inquire",
                    "If the discrete data report S6F3, F9, F11, F13 can involve more than one block,
this transaction must precede the transmission."],
                "block":"single","direction":"h<-e","reply":true,
                "autoReply":{"header":{"stream":6,"function":6,"reply":false},"body":{"format":"B","
value":0}},
                "body":["DATAID","DATALENGTH"]},

            "S6F6":{
                "doc":["Multi-block Grant"],
                "block":"single","direction":"h->e",
                "body":"GRANT6"},

            "S6F11":{
                "doc":["Event Report Send",
                        "Sends a report when an event occurs.",
                    "The purpose of this message is for the equipment to send a defined, linked, and
enabled group of reports to the host upon the",
                    "occurrence of an event (CEID).",
                    "If S6F11 is Multi-block, it must be preceded by the S6,F5/S6,F6 Inquire/Grant
transaction.",
                    "Exception: If there are no reports linked to the event a 'null' report is assumed.",
                    "A zero-length list for number of reports means there are no reports linked to the
given CEID."],
                "block":"multiple","direction":"h<-e","reply":true,
                "CommonID":["DATAID"],"realtime":true,"SQLTags":["DATAID","CEID"],
                "autoReply":{"header":{"stream":6,"function":12,"reply":false},"body":{"format":"B","
value":0}},
                "body":["DATAID","CEID",[["RPTID",["V","..."]],"..."]]},

            "S6F12":{
                "doc":["Event Report Acknowledge"],
                "block":"single","direction":"h->e",
                "body":"ACKC6"},

            "S6F15":{
                "doc":["Event Report Request",
                    "The purpose of this message is for the host to demand a given report group from the
equipment."],
                "block":"single","direction":"h->e","reply":true,
                "body":"CEID"},

            "S6F16":{
                "doc":["Event Report Data",
                    "Equipment sends reports linked to given CEID to host.",
                    "Identical to structure of S6F11.",
                    "A zero-length list for number of reports means there are no reports linked to the
given CEID."],
                 "block":"multiple","direction":"h<-e",
                "body":["DATAID","CEID",[["RPTID",["V","..."]],"..."]]},

            "S6F19":{
                "doc":["Individual Report Request",
                    "The purpose of this message is for the host to request a defined report from the
equipment."],
                "block":"single","direction":"h->e","reply":true,
                "body":"RPTID"},
            "S6F20":{
                "doc":["Individual Report Data",
                    "Equipment sends variable data defined for the given RPTID to the host."],
                "block":"multiple","direction":"h<-e",
                "body":["V","..."]}},
        "S7":{
            "doc":["Stream 7 Process Program Management",
                "The functions in this stream are used to manage and transfer process programs.",
                "Process programs are the equipment-specific descriptions that determine the procedure
to be",
                "conducted on the material by a single piece of equipment.",
                "Methods are provided to transfer programs as well as establish the link between the
process program",
                "and the material to be processed with that program."],

            "S7F0":{
                "doc":["Abort Transaction"],
                "block":"single","direction":"h<->e"},

            "S7F1":{
                "doc":["Process Program Load Inquire",
                    "This message is used to initiate the transfer of a process program or to select
from stored programs.",
                    "The message may be used to initiate the transfer of an unformatted process program
(S7F3/S7F4)",
                    "or a formatted process program (S7F23/S7F24), (S7F31/S7F32)."],
                "block":"single","direction":"h<->e","reply":true,
                "body":["PPID","LENGTH"]},

            "S7F2":{
                "doc":["Process Program Load Grant",
                    "This message gives permission for the process program to be loaded."],
                "block":"single","directions":"h<->e",
                "body":"PPGNT"},

            "S7F3":{
                "doc":["Process Program Send",
                    "The program is sent.",
                    "If S7F3 is multi-block, it must be preceded by the S7F1/S7F2 Inquire/Grant
transaction."],
                "block":"multiple","direction":"h<->e","reply":true,
                "body":["PPID","PPBODY"]},

            "S7F4":{
                "doc":["Process Program Acknowledge"],
                "block":"single","direction":"h<->e",
                "body":"ACKC7"},

            "S7F5":{
                "doc":["Process Program Request",
                    "This message is used to request the transfer of a process program."],
                "block":"single","direction":"h<->e","reply":true,
                "body":"PPID"},

            "S7F6":{
                "doc":["Process Program Data",
                    "Exception: A zero-length list means request denied."],
                "block":"multiple","direction":"h<->e",
                "body":{"or":[["PPID","PPBODY"],[]]}},

            "S7F17":{
                "doc":["Delete Process Program Send",
                    "This message is used by the host to request the equipment to delete process
programs from equipment storage.",
                    "Exception: If variable list size = 0, delete all."],
                "block":"single","direction":"h->e","reply":true,
                "body":["PPID","..."]},

            "S7F18":{
                "doc":["Delete Process Program Acknowledge"],
                "block":"single","direction":"h<-e",
                "body":"ACKC7"},

            "S7F19":{
                "doc":["Current EPPD Request",
                    "This message is used to request the transmission of the current equipment process
program directory (EPPD).",
                    "This is a list of all the PPIDs of the process programs stored in the equipment."],
                "block":"single","direction":"h->e","reply":true},

            "S7F20":{
                "doc":["Current EPPD Data",
                    "This message is used to transmit the current EPPD."],
                "block":"multiple","direction":"h<-e",
                "body":["PPID","..."]},

            "S7F23":{
                "doc":["Formatted Process Program Send",
                    "This message allows movement of formatted process programs between a piece of
equipment and its host system.",
                    "The values of MDLN and SOFTREV are obtained from the PCD used to generate the
process program.",
                    "If S7F23 is multi-block, it must be preceded by the S7F1/F2 Inquire/Grant
transaction."],
                "block":"multiple","direction":"h<->e","reply":true,
                "body":["PPID","MDLN","SOFTREV",[["CCODE",["PPARM","..."]],"..."]]},

            "S7F24":{
                "doc":["Formatted Process Program Acknowledge",
                    "Acknowledges reception of a formatted process program at its destination and
whether the process program",
                    "was accepted by the interpreter.",
                    "A returned status of \"accepted\" by the interpreter means only that the message is
understood.",
                    "The validity of the contents of the process program is determined through a
separate transaction (S7F27/S7F28)."],
                "block":"single","direction":"h<->e",
                "body":"ACKC7"},

            "S7F25":{
                "doc":["Formatted Process Program Request",
                    "This message is used by either equipment or host to request a particular process
program from the other."],
                "block":"single","direction":"h<->e","reply":true,
                "body":"PPID"},

            "S7F26":{
                "doc":["Formatted Process Program Data",
                    "This message transfers a process program in response to a request for the PPID.",
                    "The values of MDLN and SOFTREV are obtained from the PCD used to generate the
process program.",
                    "Exception: A zero length list indicates the request was denied."],
                "block":"multiple","direction":"h<->e",
                "body":["PPID","MDLN","SOFTREV",[["CCODE",["PPARM","..."]],"..."]]}},

        "S9":{
            "doc":["Stream 9 System Errors",
                "This stream provides a method of informing the host that a message block has been
received",
                "which cannot be handled or that a timeout on a transaction (receive) timer has
occurred.",
                "The messages indicate either a Message Fault or a Communications Fault has occurred
but",
                "do not indicate a Communications Failure has occurred."],

            "S9F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S9F1":{
                "doc":["Unrecognized Device ID",
                    "The device ID in the message header did not correspond to any known device ID in
the node detecting the error."],
                "block":"single","direction":"h<-e",
                "body":"MHEAD"},

            "S9F3":{
                "doc":["Unrecognized Stream Type",
                    "The equipment does not recognize the stream in the message header."],
                "block":"single","direction":"h<-e",
                "body":"MHEAD"},

            "S9F5":{
                "doc":["Unrecognized Function Type",
                    "The equipment does not recognize the function in the message header."],
                "block":"single","direction":"h<-e",
                "body":"MHEAD"},

            "S9F7":{
                "doc":["Illegal Data",
                    "This message indicates that the stream and function were recognized, but the
message format was incorrect."],
                "block":"single","direction":"h<-e",
                "body":"MHEAD"},

            "S9F9":{
                "doc":["Transaction Timer Timeout",
                    "This message indicates that the host failed to respond to a request within the
expected amount of time. The transaction has been aborted.",
                    "It is up to the host to respond to this error in an appropriate manner to keep the
system operational."],
                "block":"single","direction":"h<-e",
                "body":"SHEAD"},

            "S9F11":{
                "doc":["Data Too Long",
                    "This message to the host indicates that the equipment has been sent more data than
it can handle."],
                "block":"single","direction":"h<-e",
                "body":"MHEAD"},

            "S9F13":{
                "doc":["Conversation Timeout",
                    "Data were expected but none were received within a reasonable length of time.
Resources have been cleared."],
                "block":"single","direction":"h<-e",
                "body":["MEXP","EDID"]}},

        "S10":{
            "doc":["Stream 10 Terminal Services",
                    "The functions of this stream is to pass textual messages between operator terminals
attached to",
                    "processing and/or testing equipment and the host. The equipment makes no attempt to
interpret",
                    "the text of the message, but merely passes it from terminal keyboard to the host or
from the host to the display of the terminal.",
                    "Management of human response times to information displayed on terminals is the
responsibility of the host."],

            "S10F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S10F3":{
                "doc":["Terminal Display, Single"],
                "block":"single","direction":"h->e","reply":true,
                "body":["TID","TEXT"]},

            "S10F4":{
                "doc":["Terminal Display, Single Acknowledge"],
                "block":"single","direction":"h<-e",
                "body":"ACKC10"}},



        "S14":{
            "doc":["Stream 14 Object Services",
                "The functions in this stream are used for generic functions concerning objects,",
                "including obtaining information about objects and setting values for an object."],
            "S14F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S14F9":{
                "doc":["Create Object Request",
                    "This message is used to request an object owner to create an object instance.",
                    "OBJSPEC specifies the object owner.",
                    "Exception: If OBJSPEC is a null-length item, no object specifier is provided.",
                    "If variable list size = 0, no specific attribute settings are requested for the new
object."],
                "block":"multiple","direction":"h<->e","reply":true,
                "body":["OBJSPEC","OBJTYPE",[["ATTRID","ATTRDATA"],"..."]]},

            "S14F10":{
                "doc":["Create Object Acknowledge",
                    "This message is used to acknowledge the success or failure of creating the new
object specified.",
                    "If successful, OBJSPEC is the object specifier of the new object.",
                    "The list of attributes returned is dependent upon the type of object specified.",
                    "Exception: If OBJSPEC is a null-length item, no object was created.",
                    "If the attribute list size = 0, then no attributes of the new object are
returned.",
                    "If the errors list size = 0, no errors were detected."],
                "block":"multiple","direction":"h<->e",
                "body":["OBJSPEC",
                        [["ATTRID","ATTRDATA"],"..."],
                        ["OBJACK",[["ERRCODE","ERRTEXT"],"..."]]]}},

        "S16":{
            "doc":["Stream 16 Processing Management",
                "This stream provides protocol for a set of messages that enable the control of material
processing at equipment and equipment resources."],

            "S16F0":{
                "doc":["Abort Transaction",
                    "Used in lieu of an expected reply to abort a transaction.",
                    "Function 0 is defined in every stream and has the same meaning in every stream."],
                "block":"single","direction":"h<->e"},

            "S16F1":{
                "doc":["Multi-block Process Job Data Inquire",
                    "If any of Processing Management messages are larger than one block, then this
transaction must precede that message."],
                "block":"single","direction":"h->e","reply":true,
                "body":["DATAID","DATALENGTH"]},

            "S16F2":{
                "doc":["Multi-block Process Job Data Grant",
                    "Message to indicate if permission is granted to transmit a multi-block Job Data
message."],
                "block":"single","direction":"h<-e",
                "body":"GRANT"},

            "S16F15":{
                "doc":["PRJobMultiCreate",
                    "Use this single message to Create Multiple Process Jobs, each of which may be
unique in its association of material to process recipe.",
                    "If multi-block, this message must be preceded by the S16F1/F2 transaction.",
                    "Exception: The list for specifying material is empty when no material is specified
for the process job.",
                    "The form of data for item 3 depends on the value in MF."],
                "block":"multiple","direction":"h->e","reply":true,
                "body":["DATAID",[["PRJOBID","MF",{"or":[[["CARRIERID",["SLOTID","..."]],"..."],
                                                        ["MID","..."]]},["PRRECIPEMETHOD","RCPSPEC",
[["RCPPARNM","RCPPARVAL"],"..."]],"PRPROCESSSTART","PRPAUSEEVENT"]]]},

            "S16F16":{
                "doc":["PRJobMultiCreate Acknowledge",
                    "This message acknowledges the request and reports any errors in the creation of a
process job.",
                    "ERRTEXT contains the identifier of process jobs that were not created.",
                    "Exception: If the list of errors is empty then no errors exist."],
                "block":"single","direction":"h<-e",
                "body":[["PRJOBID","..."],["ACKA",[["ERRCODE","ERRTEXT"],"..."]]]},

            "S16F17":{
                "doc":["RJobDequeue",
                    "Used to remove process jobs from the equipment for jobs that have not begun
processing.",
                    "Exception: If the list size = 0, then de-queue all."],
                "block":"single","direction":"h->e","reply":true,
                "body":["PRJOBID","..."]},

            "S16F18":{
                "doc":["PRJobDequeue Acknowledge",
                    "Acknowledge the request to de-queue and report any errors.",
                    "ERRTEXT will contain the identifier of any jobs that were not de-queued.",
                    "Exception: If error list size = 0, no errors exist."],
                "block":"single","direction":"h<-e",
                "body":[["PRJOBID","..."],["ACKA",[["ERRCODE","ERRTEXT"],"..."]]]},

            "S16F19":{
                "doc":["PRGetAllJobs",
                    "Requests the equipment to return a list of process jobs which have not completed.",
                    "They may be running or waiting to run."],
                "block":"single","direction":"h->e","reply":true},

            "S16F20":{
                "doc":["PRGetAllJobs Send",
                    "Returns the requested list of process jobs.",
                    "Exception: If the list of jobs = 0, then no process jobs are running or waiting to
run."],
                "block":"single","direction":"h<-e",
                "body":[["PRJOBID","PRSTATE"],"..."]}}
        }
}
```

SECS/GEM Messages

SECS Message Basics

From the SECS-II standard:

> "SECS-II defines the method of conveying information between equipment and host in the form of messages. These messages are organized into categories of activities, called streams, which contain specific messages, called functions. "

Each SECS message specifies a stream and a function. Specific SECS messages are often referred to in this format: **S**(number)**F**(number). For example **S1F13** refers to SECS message Stream 1, Function 13.

Requests and Responses

There are two kinds of messages:

- **Request**: sometimes called the **primary message**. These originate from the **Host**, which in the context of Ignition, is the Ignition Gateway. Requests are odd-number functions, such as S1F1.
- **Response**: sometimes known as a **secondary message**. These originate from the **Equip ment**. Responses are even-number functions, such as S1F2

Thus, the Host can send an S1F1 Request, which establishes if the Equipment is on-line. If so, then the Equipment will respond with an S1F2 Response, which signifies that the Equipment is on-line. Each request can only have a single response message, or no response message if none is needed. This Request-and-Response pattern is the basis of the module, and the standard.

On this page ...

SECS Message Format

Every SECS message is a JSON formatted string, which contains a header. The header contains the stream, function, and if the message expects a response message. Optionally, the SECS message can have a body, which consists of a list of one or more data items. Each data item can also be a list of other data items. This forms the basis for which all SECS messages should look like, regardless of if they are a request or response.

SECS Message Example

The following is an example of a S1F11 request:

S1F11 Request Message

```
{
   "header":{
      "doc":"Status Variable Namelist Request",
      "stream":1,
      "function":11,
      "reply":true
   },
   "body":[
      {
         "doc":"SVID, Status Variable ID",
         "format":"U4",
         "value":4
      }
   ]
}
```

When making this request from a script, the sendRequest function takes in the header values as parameters, with the whole body as another parameter.

Python - Send an S1F11 request to the Equipment

```
body = [{"format":"U4", "value":4}]

transactionID = system.secsgem.sendRequest("S1F11",True,body,"Equip4")
```

A typical response to the above request would look something like this:

S1F12 Response Message

```
{
   "header":{
      "doc":"Status Variable Namelist Reply",
      "stream":1,
      "function":12,
      "reply":false
   },
   "body":[
      [
         {
            "doc":"SVID, Status Variable ID",
            "format":"U4",
            "value":4
         },
         {
            "doc":"SVNAME, Status Variable Name",
            "format":"A",
            "value":"RandomBinaryData"
         },
         {
            "doc":"UNITS, Units Identifier",
            "format":"A",
            "value":"B"
         }
      ]
   ]
}
```

Each response message would be received using the getResponse function, which would use the transactionID from the sendRequest call above, and then do something with the response such as printing it.

Python - Get an S1F12 response from the Equipment

```
response = system.secsgem.getResponse(transactionID, "Equip4")

print response
```

This simple request/response between the Gateway and Equipment forms the basis for communication using the SECS/GEM module. However, there are some other tools that allow you to receive the response message in different ways: through Tags and Message Handlers.

In order to send a certain function request, the equipment connection must have that function defined in the SDL File.

SECS Messages through Tags

Any SECS messages that contain data can be captured in Tags that can then be used within a project. This can be really useful for something like data coming back from tool traces within the S6F1 messages. When properly configured, the system will automatically create the appropriate Tags within the equipment's folder in the SECSGEM Realtime Tag Provider. While the Tags that get created can be bound to components, they cannot be written to and they can't be edited which means no Tag History, no Alarming, etc. Also note that if the Gateway is ever restarted those Tags will disappear, so they will need to be recreated. To have responses generate Tags, the function must be configured to do so within the SDL File. This is done by adding an object called 'SQLTags' with a value of 'true', along with a 'CommonID' array that holds an identifier. The default SDL file is preconfigured to create Tags for S6F1 messages. Sending a S2F23 request with the appropriate trace parameters to the tool will produced Tags based on the response message. With each new S6F1 message that is received, the Tags will update their values. Using the built in simulator, you can easily create Tag values by calling the S2F23 request with the body below.

Body of an S2F23 request

```
[
   {
      "doc":"TRID, Trace Request ID.",
      "format":"U4",
      "value":1
   },
   {
      "doc":"DSPER, Data sample period.",
      "format":"A",
      "value":"000002"
   },
   {
      "doc":"TOTSMP, Total Samples",
      "format":"U4",
      "value":10
   },
   {
      "doc":"REPGSZ, Reporting Group Size",
      "format":"U4",
      "value":1
   },
   [
      {
         "doc":"SVID Status Variable ID",
         "value":1,
         "format":"U4"
      },
      {
         "doc":"SVID Status Variable ID",
         "value":2,
         "format":"U4"
      },
      {
         "doc":"SVID Status Variable ID",
         "value":3,
         "format":"U4"
      }
   ]
]
```

Example with S2F14 Using the Simulator

This example sets up the S2F14 message to create Tags that we can see in the Tag provider. It assumes the equipment connection is made to a simulator with the default SDL file. In order to capture other messages in Tags, the SDL File will need to be configured to allow that. Here, we setup the SDL File to have S2F14 to create Tags.

1. Go to the Gateway Webpage Configure section, and navigate to SECSGEM -> Equipment.
    a. On the equipment connection, edit the SDL File.
2. Locate the S2F14 message in the list.
    a. Add "SQLTags" as an Auto type object with a value of "true" directly under the S2F14 root.
    b. Create an array called "CommonID" directly under the S2F14 root.
        i. Add "0: ECV" to the CommonID array as an Auto type object.
3. Save the SDL file and restart the equipment connection.
4. Once the equipment connection has been reestablished, send it an S2F13 message with the body below:

S2F13 Body

```
[
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":200
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":201
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":203
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":205
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":206
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":207
   }
]
```

SECS Messages through Message Handlers

Similar to how SECS message data can be captured in Tags, they can instead be made to call a custom Python script. To implement a custom Python script, the message will need to call a Message Handler with the name **onSecsGemRealtimeUpdate**.

Message Handler Limitations

The Message Handler can only work with message that are allowed to originate on equipment, such as S6F1.

In cases where the message can only be sent from the host, such as S2F33, then the Message Handlers will be unable to respond.

This Message Handler must have that exact name, though a handler can be specified for both the Client Event Scripts and the Gateway Event Scripts. Within the Message Handler, the SECS message will pass in information into the payload object.

Payload Key	Value
CommonID	The common ID of the message.
Equipment	The name of the equipment that the message is coming from.
Direction	The direction of the message.

RequestResponse	If this was from a request or a response.
TxID	The transaction ID of the message.
Reply	If this message expected a reply.
Message	The message contents.

SECS/GEM Message Handler Example

This script can handle the message in any way such as storing data in the database or writing to Tags. The example below will take the information and create a logger object with it that will then be printed to the console.

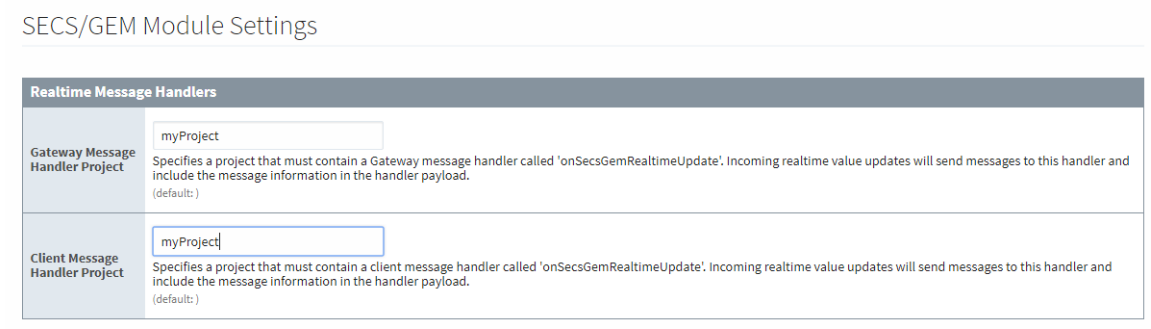
Python - Example of a onSecsGemRealtimeUpdate Message Handler

```
# This message handler is very simple in that it will create a logger with the SECS message data that
prints to the console.

commonID = payload['CommonID']
equipment= payload['Equipment']
direction = payload['Direction']
reqResp = payload['RequestResponse']
txId = payload['TxID']
reply = payload['Reply']
message = payload['Message']

msg = "CommonID=" + commonID
msg += ", Equipment=" + equipment
msg += ", Direction=" + direction
msg += ", RequestResponse=" + reqResp
msg += ", TxID=" + str(txId)
msg += ", Reply=" + str(reply)
msg += ", Message=" + message
logger = system.util.getLogger("SECSGEM.Gateway.onSecsGemRealtimeUpdate")
logger.info(msg)
```

Once an onSecsGemRealtimeUpdate Message Handler has been specified, it needs to be specified within the Gateway. Within the Gateway Webpage Configure section, the **Module Settings** page under the SECS/GEM header will have Properties to specify which project holds the Gateway Message Handler and which project holds the Client Message Handler. These can be the same project or different projects.


SECS/GEM Module Settings

Realtime Message Handlers	
Gateway Message Handler Project	myProject Specifies a project that must contain a Gateway message handler called 'onSecsGemRealtimeUpdate'. Incoming realtime value updates will send messages to this handler and include the message information in the handler payload. (default:)
Client Message Handler Project	myProject Specifies a project that must contain a client message handler called 'onSecsGemRealtimeUpdate'. Incoming realtime value updates will send messages to this handler and include the message information in the handler payload. (default:)

Example with S2F14 Using the Simulator

This example sets up the S2F14 message to call the Gateway Message Handler that we setup. It assumes the equipment connection is made to a simulator with the default SDL file. In order to capture message with the onSecsGemRealtimeUpdate, the SDL File will need to be configured to allow that. Here we setup the SDL File to have S2F14 to go through the message handler.

1. Create a Gateway Message Handler called **onSecsGemRealtimeUpdate**.
2. Go to the Gateway Webpage Configure section, and navigate to the SECS/GEM ->Module Settings page.
    a. Specify the project that the Gateway Message Handler was created on in the Gateway Message Handler Project setting.
3. Within the Gateway Webpage Configure section, navigate to SECS/GEM -> Equipment.
    a. On the equipment connection, edit the SDL File.
4. Locate the S2F14 message in the list.
    a. Add "realtime" as an Auto type object with a value of "true" directly under the S2F14 root.
    b. Create an array called "CommonID" directly under the S2F14 root.
        i. Add "0: ECV" to the CommonID array as an Auto type object.

5. Save the SDL file and restart the equipment connection.
6. Once the equipment connection has been reestablished, send it an S2F13 message with the body below:

S2F13 Body

```
[
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":200
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":201
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":203
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":205
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":206
   },
   {
      "doc":"ECID, Equipment Constant ID",
      "format":"U4",
      "value":207
   }
]
```

Custom SECS Message Response Handlers

For messages that expect a response, the system is setup to automatically send a response based on what is configured in the SDL file. However, a custom response can be created using a Message Response Handler, allowing you to build custom responses completely in the Python scripting language. Only one custom response to a stream function can be configured per equipment connection, but each equipment connection can have multiple custom responses, each to a different stream function. By setting up a custom Message Response Handler for a stream function, the handler will be used for all responses to the specified stream function.

A Message Response Handler is setup on an equipment connection and pointed at a Gateway Event Script Message Handler. Each Message Handler will have a payload object that will contain information from the default message response.

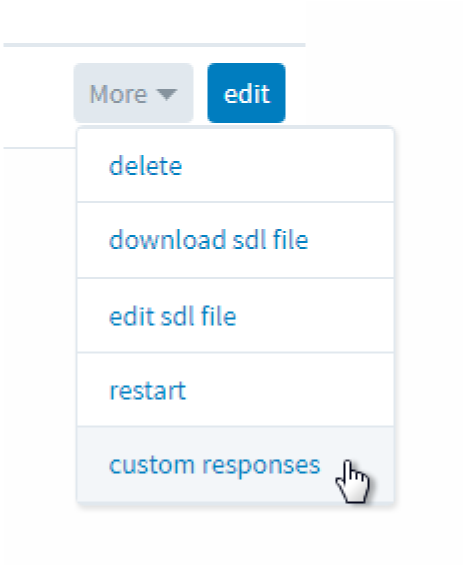
Payload Key	Value
Equipment	The name of the equipment that the message is coming from.
TxID	The transaction ID of the message.
SystemBytes	The system bytes of the response.
Message	The message contents.

Using the data given to us in the payload, the Message Handler can then manipulate it in any way before calling the system.secsgem.sendResponse function, which will send the response back to the equipment. The response from the custom handler only execute when receiving messages over the HSMS protocol.

The following feature is new in Ignition version **8.0.16**
Click here to check out the other new features

As of 8.0.16, custom response handlers will execute when receiving messages over the SECS-1 (serial) protocol.

Once a Message Handler has been created, the equipment connection needs to specify that it will be using that as a custom message response. To do this, navigate to the Gateway Webpage Configure section, navigate to SECS/GEM -> Equipment. For the equipment connection, click the more button on the right and select the Custom Responses option



On the Message Response Handlers page, we can setup a link between the stream function response we want to intercept and which Message Handler will be used to handle it. Each Message Response Handler has the following settings.

Setting	Description
Enabled	If true, the specified script message handler will be fired when the specified StreamFunction is encountered.
Intercept Stream Function	The SECS stream and function type of the message that will be intercepted. Use a value such as "S6F2"
Gateway Message Handler	The name of the script message handler that will create the SECS response message. This message handler must be created in a project to provide custom responses, and must be a Gateway Event Script message handler.
Project Name	The project that contains the script message handler specified above.

SECS/GEM Simulator

Overview

The SECS/GEM Module includes the capability to add and configure equipment simulators. Equipment simulators include basic functionality and have the ability to handle and respond to a number of SECS messages. You can add status variables, equipment constants, reports and collection events to a simulator, allowing you to model a simulator after an existing tool. You can also add hard-coded responses to messages that do not exist in the simulator. These functions allow the simulator to act as a stand-in for tools that do not provide any type of emulator, or only provide one at great cost.

On this page ...

Simulator SDL File

A simulator is driven from data in its SDL file. This file is the same format as the SDL file used for equipment connections, but it also defines variables, reports, collection events, and hard-coded responses. The simulator SDL file is read when a simulator is started.

You can modify the simulator SDL file within the Gateway. Navigate to Configuration area and click on the SECS/GEM Simulator link. Click on the "sim variables" link next to the simulator you want to modify. The SDL file will be displayed in a tree, allowing you to browse its contents. You can change individual values on an object in the tree. Click the "Save SDL File" button at the top to save any changes you made. You can also click the "Edit Raw SDL File" link to edit the text of SDL file directly. This can be handy when you need to change several objects at once, or see a full view of the SDL file. Click the "Save SDL File" button at the top to save any changes you made. Don't forget to restart the simulator and its equipment connection after you made changes to the SDL file.

Simulator SDL Utilities

Modifying the simulator SDL file works well for small adjustments, but attempting to create variables, reports and collection events from scratch directly within the SDL file is extremely painful. To eliminate this pain, the Simulator Variables page has a number of utilities that can automate the creation of these objects. Note that after these objects are created, you will still need to modify the simulator SDL file directly to modify or delete the objects.

Default Messages

By default, the Simulator can respond to the following requests.

- **S1F1** - Are You Online
- **S1F3** - Selected Equipment Status Request
- **S1F11** - Status Variable Namelist Request
- **S1F13** - Establish Communications Request
- **S1F15** - Request OFF-LINE
- **S1F17** - Request ON-LINE
- **S2F13** - Equipment Constant Request
- **S2F15** - New Equipment Constant Send
- **S2F17** - Data and Time Request
- **S2F23** - Trace Initialize Send
- **S2F29** - Equipment Constant Namelist Request
- **S2F31** - Date and Time Set Request
- **S2F33** - Define Report

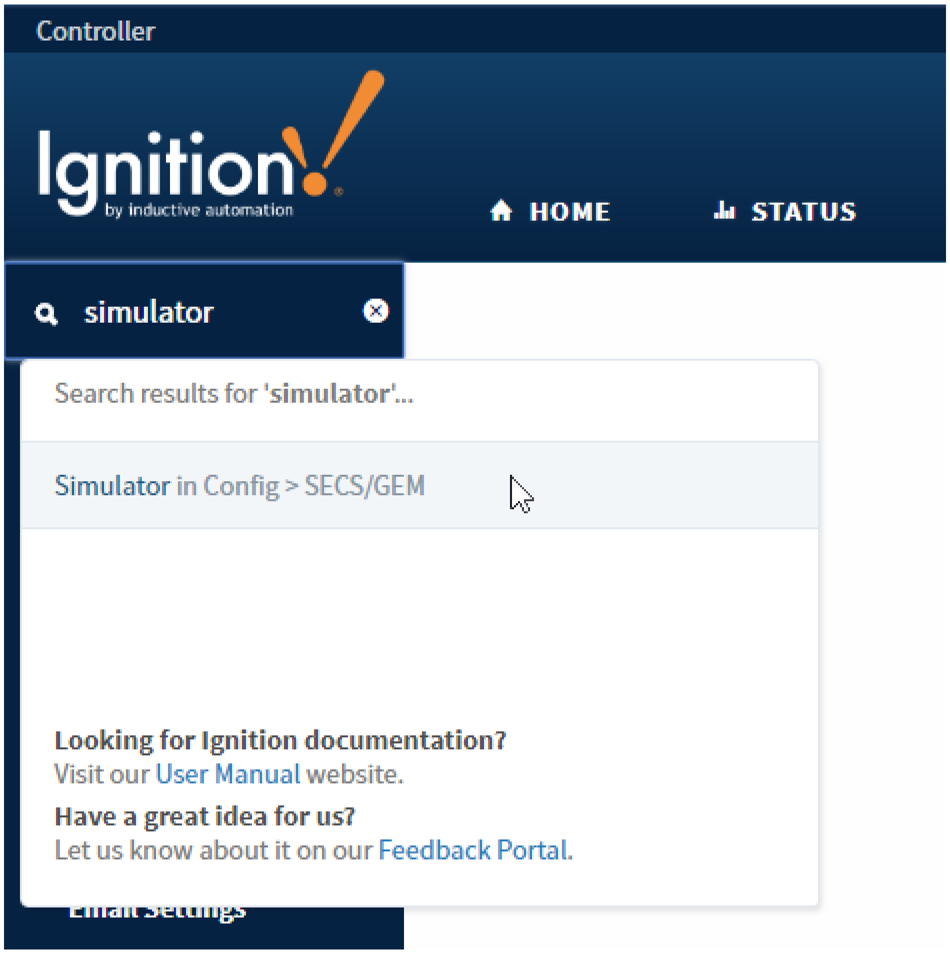
While the SDL can be modified, note that the Simulator can only respond to the requests listed above: you can add new messages to the simulator's SDL, like S999F1 and S999F2, but the Simulator by default will not respond to any requests using S999F1. If you wish to add new responses, you can create an Echo Response to simulate a response to any request not listed above. See the Echo Response section for more details.

The simulator implements the Communications State Model and the Control State Model from the GEM standard (SEMI E30). For example, sending an S1F15 message to a simulator will cause the simulator to move to the OFF-LINE state and will thereafter respond to most messages with S1F0. Sending an S1F17 message to the simulator will move it back into an online state. When control states change, an S6F11 report message is sent with the updated control state. The default startup control states are defined in the simulator SDL file.

Configuring a Simulator

The following section details how to configure a SECS/GEM simulator.

1. Navigate to the SECS/GEM Simulator page:
   - From the **Configure** section of the Gateway webpage, scroll down the list of links on the left until you see the **SECS/GEM** section. click on the **Simulator** link.
   - Alternatively, use the search bar located in the upper left corner of any page on the Gateway, and search for `simulator`.


Controller
Ignition by inductive automation
HOME
STATUS
simulator
Search results for 'simulator'...
Simulator in Config > SECS/GEM
Looking for Ignition documentation?
Visit our User Manual website.
Have a great idea for us?
Let us know about it on our Feedback Portal.
Email Settings

2. Once on the Simulators page, click on the **Create New Simulator** link.

3. From here we can create a new simulator. In the very least, you will need to specify a name for the simulator.

4. If this is the first SECS/GEM simulator device connection, then you can use the default values for the rest of the properties and skip to the next step.

   Subsequent simulators require a different port: two simulators cannot run on the same port. Note that there are multiple port properties.
   - If the **Connection Mode** is set to Active, then the simulator will use the **Active Port**
   - If the **Connection Mode** is set to Passive, then the simulator will use the **Passive Port**
   - If the **Connection Mode** is set to Alternating, then the simulator will try to use either the **Active Port** or **Passive Port**. As the name implies, the connection will attempt to alternate between the Active and Passive settings until a connection is established.

5. Click on the **Create New Simulator** button to save your configuration and create the simulator device. You will see the Simulators page again.

6. Note, that the simulator status will toggle between "Connecting..." and "Not Connected" until you create an equipment connection tha t connects to the simulator's IP address and port. Once the Equipment Connection is established, the simulator will report a status of "Communicating". You can then run SECS/GEM commands against the simulator's equipment connection and receive responses.

Simulator Properties

When configuring a simulator, there are a number of properties that can be configured.

Property Name	Description	Default
Simulator Name	Name of the simulator.	
Simulator Description	A description of the simulator.	
Enabled	Whether the simulator is enabled or disabled. A disabled simulator will prevent communication with the Equipment Connection.	true
Active IP Address	IP Address of Equipment Connection to connect to. Used when simulator is in Active mode.	localhost
Active Port	The Port number of the Equipment Connection to connect to when simulator is in Active mode. When using multiple simulators, the port numbers need to be unique.	5000
Passive IP Address	IP Address of Ignition that the Equipment Connection will connect to when the simulator is in Passive mode.	localhost
Passive Port	Port number that the Equipment Connection wil connect to when the simulator is in Passive mode. When using multiple simulators, the port numbers need to be unique.	5000
Device ID	Unique identifier of equipment. Must be an integer.	0
Connection Mode	Method used to connect. • ACTIVE- Will attempt to connect to the equipment at the given Active IP Address and Active Port. • PASSIVE - Will listen for a connection from the equipment at the given Passive IP Address and Passive Port. • ALTERNATING - Will switch between Active mode and Passive mode until a connection is made.	PASSIVE
Advanced Properties		
T3 Reply Timeout	Specifies the seconds that the simulator will wait for an expected SECS message reply.	45
T5 Connect Separation Timeout	Specifies the seconds which must elapse between successive attempts to connect to an Equipment Connection after disconnection.	10
T6 Control Transaction Timeout	Specifies the seconds which a control transaction (such as LinkTest or Select) may remain open before it is considered a communications failure.	5
T7 Not Selected Timeout	Seconds which a TCP/IP connection can remain in NOT SELECTED state (i.e., no HSMS activity) before it is considered a communciations failure. This timeout is only used in Passive mode.	10
T8 Network Intercharacter Timeout	Maximum seconds between successive bytes of a single HSMS message which may expire before it is considered a communications failure. This applies to HSMS messages received from an Equipment Connection.	5

Related Topics ...

- SECS/GEM Equipment Connections
- Using the SECS/GEM Module

In This Section ...

Additional SECS/GEM Simulator Features

Default Variables

Each simulator is initialized with a number of status variables and equipment constants for demonstration purposes. The status variables and equipment constants are "live" objects within the simulator. This means that the standard SECS/GEM commands (such as S1F3) can be run to view the variable definitions and current values. Equipment constant values can also be changed using the S2F15 command. See the "Simulator SDL file" section below for information on how to change the values directly in the simulator.

Import Status Variable Definitions

If you can capture status variable definitions from an existing tool using the S1F11 command, you can automatically create status variables within the simulator. Save your capture to a .txt file with the following format (substitute your own status variables):

```
{
   "body":[
      [
         {
            "doc":"SVID, Status Variable ID",
            "format":"U4",
            "value":268435456
         },
         {
            "doc":"SVNAME, Status Variable Name",
            "format":"A",
            "value":"LMMode"
         },
         {
            "doc":"UNITS, Units Identifier",
            "format":"A",
            "value":""
         }
      ],
      [
         {
            "doc":"SVID, Status Variable ID",
            "format":"U4",
            "value":268435457
         },
         {
            "doc":"SVNAME, Status Variable Name",
            "format":"A",
            "value":"LMState"
         },
         {
            "doc":"UNITS, Units Identifier",
            "format":"A",
            "value":""
         }
      ]
   ]
}
```

On the Simulator Variables page, click on the "Import status variable definitions" link. Choose your text file and click "Begin Import". If there were no errors during import, you will be returned to the Simulator Variables page. You can see your new status variables in the tree under data -> Status Variables or by viewing the raw SDL file. Note that all imported status variables have a format of "UNKNOWN" and a value of "UNKNOWN", due to the fact that the S1F11 command does not send status variable format or value. You must manually set these values before the status variables can be read by the simulator. After the format and value of the new status variables are set, you can save the simulator SDL file, and restart the simulator and its equipment connection. The new status variables will now be readable within the simulator.

Import Equipment Constant Definitions

If you can capture equipment constant definitions from an existing tool using the S2F29 command, you can automatically create equipment constants within the simulator. Save your capture to a .txt file with the following format (substitute your own equipment constants):

```
{
   "body":[
      [
         {
            "doc":"SVID, Status Variable ID",
            "format":"U4",
            "value":268435456
         },
         {
            "doc":"SVNAME, Status Variable Name",
            "format":"A",
            "value":"LMMode"
         },
         {
            "doc":"UNITS, Units Identifier",
            "format":"A",
            "value":""
         }
      ],
      [
         {
            "doc":"SVID, Status Variable ID",
            "format":"U4",
            "value":268435457
         },
         {
            "doc":"SVNAME, Status Variable Name",
            "format":"A",
            "value":"LMState"
         },
         {
            "doc":"UNITS, Units Identifier",
            "format":"A",
            "value":""
         }
      ]
   ]
}
```

On the Simulator Variables page, click on the "Import equipment constant definitions" link. Choose your text file and click "Begin Import". If there were no errors during import, you will be returned to the Simulator Variables page. You can see your new equipment constants in the tree under data -> Equipment Constants or by viewing the raw SDL file. Restart the simulator and its equipment connection. The new equipment constants will now be readable within the simulator.

Echo Responses

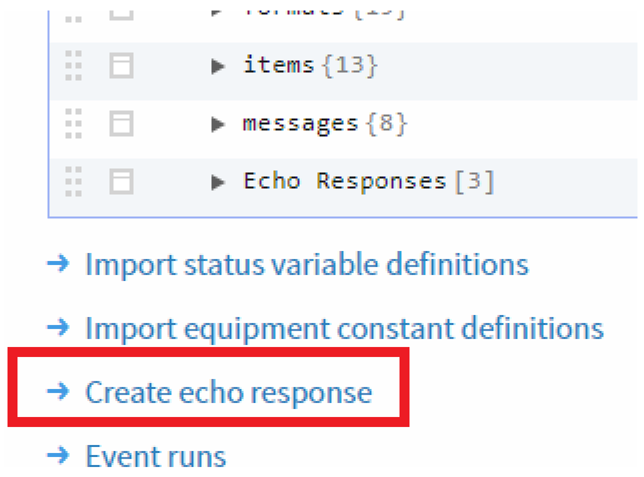
There are a number of SECS/GEM messages that are not implemented in the default simulator. However, you can still use this utility to provide a hard-coded response to these messages.

When creating an Echo Response, you are simply creating a dedicated response to a specified request. Echo Responses can be used to override the responses contained in the Simulator's SDL.

Additionally, you can create an Echo Response to a message that isn't included in the Simulator's Default Messages, allowing you to have the Simulator emulate another tool. In these cases, it you will need to capture the output of an actual tool's response to a message or know the exact format of the response. Furthermore, you will need to modify the Simulator's SDL file to include the message both the request and response.

Creating an Echo Response

On the Simulator Variables page, click on the **Create echo response** link.


items {13}
messages {8}
Echo Responses [3]
Import status variable definitions
Import equipment constant definitions
Create echo response
Event runs

The following properties will appear:

Property Name	Description
Name	The name of the response. Used purely to help you identify specific responses when viewing the responses from the SDL.
Stream and Function	The Stream and Function that this Echo will respond to.
Echo Response	The message to return. If emulating another tool, you will want to modify the Echo Response to match an actual response from the tool.

Example

1. If you wanted to create a custom response to S1F1, specify Stream 1, Function 1, and use the Message Below

```
{
   "header":{
      "stream":1,
      "function":2,
      "reply":false,
      "doc":"On Line Data"
   },
   "body":[
      {
         "format":"A",
         "value":"SimOne",
         "doc":"MDLN, Equipment Model Type"
      },
      {
         "format":"A",
         "value":"1",
         "doc":"SOFTREV, Software Revision Code"
      }
   ]
}
```

2. Click **Save**
3. Click the **Save SDL File** button
4. Restart the Simulator.
5. Now, whenever the simulator receives an S1S1 message, it will response with your custom S1F2 message.

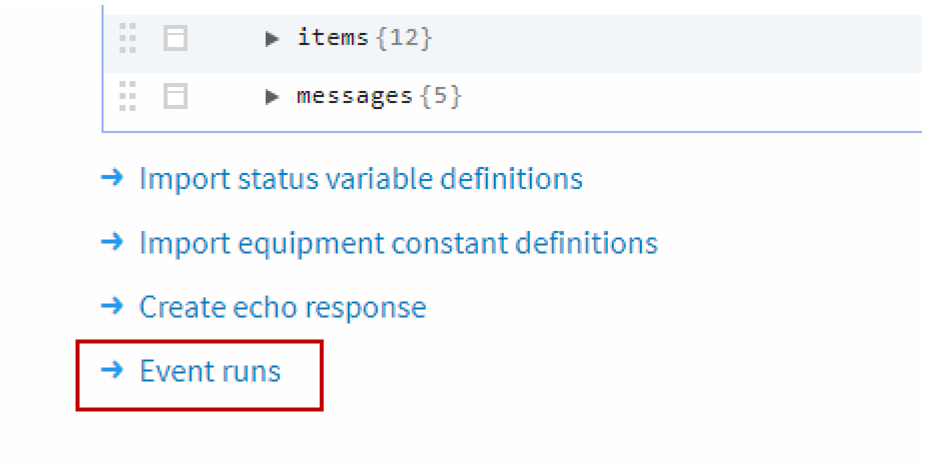
Remember: if you added a new function, you may also need to define the message in the messages section of both the simulator SDL file and the equipment connection SDL file. You may also need to add new format items to the items section of both SDL files. Failure to do so can result in "message not found" or "format not found" errors when you attempt to send the command to the simulator.

Event Runs

The Simulator features an Event Run feature, which allows you to create a list of S6F11 messages at fixed intervals that simulate report generation. An Event Run consists of one or more events, with each event containing a S6F11 message that will be sent to the simulator when the event fires.

Add Event Runs

Each simulator can have multiple event runs specified, each with a unique set of events. To access a simulators event runs, go to the Gateway Webpage Configure section and navigate to the SECS/GEM -> Simulator page, and in the **More** menu to the right of the simulator select the **sim variables** option. At the bottom of the SDL editor tool, there will be a link for **Event Runs** that will take you to the page where new event runs can be created.


items {12}
messages {5}
Import status variable definitions
Import equipment constant definitions
Create echo response
Event runs

On the Event Runs page, a new event run can be created by clicking the **Create new Event Run** link, where you can specify the name of the new event run.

Add Events

Once an event run has been made, new events can be added to it that will execute in order from when the event was started. An event run can have any number of events within it. Events must be from S6F11 messages that have been previously captured, which can be copied from the Messages table in the database. To add a new event, navigate to the Event Runs page and click on the **More** menu to the right of the event run that you wish to add events to. Select the **events** option to navigate to the Events page where you can click on the **Create new Event** link to create a new event in that event run. Each event has a few different properties that can be configured.

Property Name	Description
Event Name	The name of the event.
Index	The index of the event. Note, each event must have a different index in the event run.
Delay	The delay in seconds from the last event until this event fires.
Send Data	A JSON formatted message that will be sent to the equipment connection when the event fires. The message must include header and body fields.

Below is an example of a message for the Send Data property.

JSON - A Sample Send Data Message for an Event Run

```json
{
   "header":{
      "stream":6,
      "function":11,
      "reply":true,
      "doc":"Event Report Send"
   },
   "body":[
      {
         "format":"U4",
         "value":2,
         "doc":"DATAID, Data ID"
      },
      {
         "format":"U4",
         "value":3,
         "doc":"CEID, Collected Event ID"
      },
      [
         [
            {
               "format":"U4",
               "value":1,
               "doc":"RPTID, Report ID"
            },
            [
               {
                  "format":"A",
                  "value":"150408103706",
                  "doc":"V, Variable data"
               },
               {
                  "format":"A",
                  "value":"OFF-LINE/HOST OFF-LINE",
                  "doc":"V, Variable data"
               }
            ]
         ]
      ]
   ]
}
```

Using Event Runs

To use an event run, it needs to be started. To start an event run navigate to the Event Runs page, and in the **More** menu to the right of the event run you wish to start select the **start** option. If at any time you wish to cancel the event run, simply select the **cancel** option from the **Mo re** menu.

Import Events

Instead of manually entering in events, multiple events can be imported from previously captured messages in the Messages table. Simply click on the **Import Event**s link on the Events page. This will allow you to set up an import with various parameters.

Import Parameter	Description
Datasource	The datasource to pull events from.
Equipment Name	The name of the equipment connection that the events are coming from.
Equipment Table Prefix	The prefix of the specified equipment's database tables.
Range Start	The starting date and time to import events from.
Range End	The ending date and time to import events from.

Once the import parameters have been set up, clicking Begin Import at the bottom will start the import process, which will pull any possible events based on the specified parameters.

Basic SECS/GEM Troubleshooting

Equipment Connections

On Equipment Connection Startup Database Tables Cannot be Created

If you are using MySQL, make sure the latest MySQL JDBC driver is installed in Ignition. If not, you can download it from the MySQL website at http://dev.mysql.com/downloads/connector/j/, or from the **Config> Databases > Drivers** section of the Gateway Webpage.

Unable to Establish Equipment Connection at Setup

If you are unable to create an equipment connection after configuring the connection settings, check the Active and Passive Port numbers. Verify that these port numbers are not being used by another piece of equipment or simulator. Everytime you configure another piece of equipment, the Active and Passive Ports numbers must be unique. They cannot be the same as another piece of existing equipment.

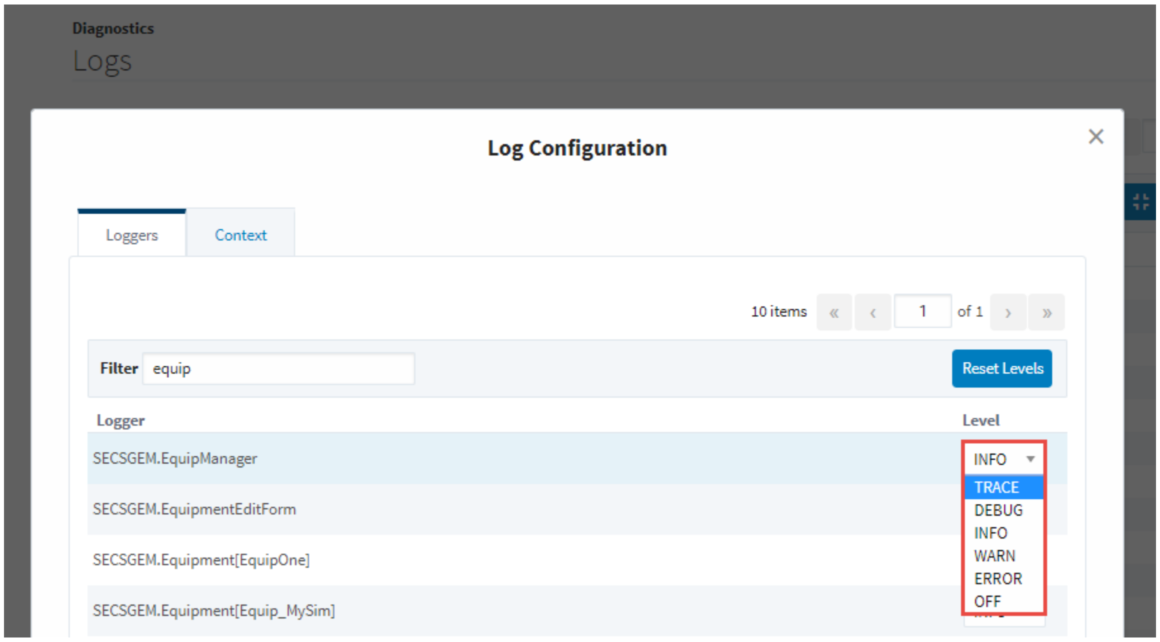
To check the port numbers, go to the **Config > SECS/GEM Equipment** section of the Gateway Webpage. Locate your equipment name, and open the **Equipment Connection** page. Verify that the Active and Passive Port numbers are not being used by another piece of equipment. If so, change the Active and Passive Ports to a unique number, then save your changes. The Equipment Connections page will update, and your equipment will have a status of "Communicating."

On this page ...

Logging

Each Equipment Connection has a logger associated with it. The Equipment Connection loggers and their severity level can be seen in the **St atus** section of the Gateway Webpage under **Diagnostics > Logs**. By default, only messages with a severity level of INFO or more severe such as WARNING or ERROR are logged. A lot of data about what an Equipment Connection is doing can be logged by changing the log level of the Equipment Connection to a less severe level. At the TRACE log level, all SECS messages sent and received by an Equipment Connection are logged.

To change the severity level of a logger, click the **Settings** ⚙ button to open the Log Configuration window. Locate the logger you want change, and click on the dropdown arrow on the right side of the window. Select the severity you want to update.



Logger	Level
SECSGEM.EquipManager	INFO
SECSGEM.EquipmentEditForm	
SECSGEM.Equipment[EquipOne]	
SECSGEM.Equipment[Equip_MySim]	

Wrapper.log File

Messages and log events can be viewed from Ignition's log file called **"wrapper.log"** as well as the Status section of the Gateway Webpage under **Diagnostic > Logs**. If something is not working as expected, it can be very useful to turn on ALL logging for an Equipment Connection and view the wrapper.log file. Often an exception error or log message will explain what is happening.

The file path location of the wrapper.log file is different for each operating system:

- **Windows** - Program Files\ Inductive Automation\Ignition\logs
- **Lunix** - /var/log/ignition
- **Mac OS X** - /user/local/ignition/logs

When logging data on Linux it can be useful to run the command tail **-f /var/log/ignition/wrapper.log**. This command displays current log messages on the screen as they occur. A similar tail program can be used on Windows.

Failure to Send a SECS Message

All SECS messages that are sent and received are validated against the SECS Definition Language (SDL) file. A common reason for failure to send and receive a SECS message is it is either not defined or doesn't match a definition in the SDL file. Any messages that fail to validate against the SDL file will be inserted into the Errors database table for the equipment connection as a validation error along with information about why it didn't validate. If either a sent or received message occurs, a log message describing the problem is written to the wrapper.log file, as well as displayed in the Diagnostic Logs for the equipment connection.

To define or update a SECS message, the SDL file can be edited in the Gateway Webpage from the Equipment Connections page using the built in SDL editor, or downloaded from the Equipment Connections page so it can be edited in a text editor and re-uploaded.

Related Topics ...

- Diagnostics