



**DARPA  
TRIAGE  
CHALLENGE**

**Interface Control Document  
Phase 2  
Data Competition  
Revision 02  
January 2025**



---

**Defense Advanced Research Projects Agency**  
Biological Technologies Office  
675 North Randolph Street  
Arlington, VA 22203-2114  
[TriageChallenge@darpa.mil](mailto:TriageChallenge@darpa.mil)

Distribution Statement 'A' (Approved for Public Release, Distribution Unlimited)

# 1 Contents

1	CONTENTS .....	2
2	INTRODUCTION.....	3
3	EVALUATION SYSTEM .....	3
3.1	EVALUATION PROCESS .....	5
3.2	COMMUNICATION PROTOCOL .....	6
4	EVALUATION ENVIRONMENT .....	13
5	SUBMISSION DEVELOPMENT .....	14
5.1	REQUIREMENTS .....	14
5.2	CLIENT SHELL .....	15
5.3	EVALUATOR CONTAINER TESTING.....	15
5.4	CODEBUILD (CI/CD) COMPLIANCE TESTING .....	16
6	SUBMISSION PROCEDURE.....	16
7	APPENDIX A – FIELDS PROVIDED DURING EVALUATION .....	17

## 2 Introduction

The goal of this document is to convey the high-level concept and infrastructure used to evaluate team submissions in the Data Competition. This document also describes the requirements and interface for successful integration with JHU/APL's evaluation system. This document is specific to the Data Competition for Phase 2. For information on the Systems and Virtual Competitions please refer to their respective *Interface Control Documents*. Significant revisions from past versions in this document are indicated by [blue](#) text.

The remainder of this document is organized as follows: Section 3 describes the evaluation system, messaging protocol, and the evaluation process; Section 4 describes the evaluation environment for Phase 2; Section 5 describes requirements and resources for submission preparation; and Section 6 describes the procedure for submission. Appendices provide supporting information.

## 3 Evaluation System

This section contains information about the evaluation system planned for workshop and challenge events in the Data Competition. All formal evaluation procedures will be performed on JHU/APL networks.

Model submissions will be evaluated using a *held-out* test dataset in a simulated online prediction environment, in accordance with the scoring procedure described in the DTC Rules Document. Models will be evaluated on a single patient case at a time. For each patient case, models will be provided with sequential data segments over the course of the case. As the case unfolds, models are given the opportunity at each segment to predict future LSIs relative to the current segment timestamp. Segments typically contain 5 minutes of data, with smaller or larger windows at the edges of a case. Segments are time-ordered and non-overlapping with only new, unseen data provided in each segment. Models will be responsible for accumulating or storing past data within a case, if necessary. Segments may not be contiguous; there may be time gaps between the end of one segment and the beginning of the next in which no new data was available. As part of the input data for each segment, models will be provided with start and stop timestamps and an indicator for the end of the case (see Section 3.2 for details).

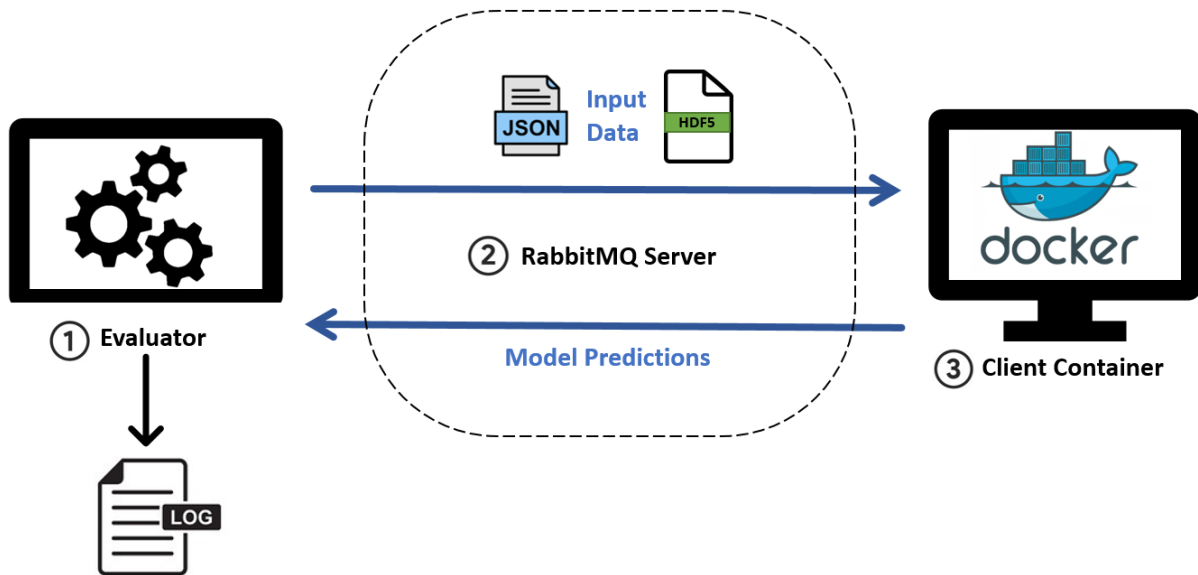
Segmented training datasets will be provided in the same form as data will appear during the evaluation. See the DTC Forum for updates to the training datasets available on AWS. The data version for Challenge Event 2 will be released in future versions of this document.

Figure 1 provides a high-level description of the three interacting modules in the JHU/APL evaluation system:

1. **Evaluator**, which hosts all logic required to distribute test data to the teams' client containers (via Rabbit MQ) and evaluate their responses;
2. **Rabbit MQ Server**, which hosts the server that governs communication between the Client Container and the Evaluator;
3. **Client Container**, a Docker container which processes input data, runs model inference, and responds with LSI predictions (via Rabbit MQ). Also referred to simply as the Client.

For each competition event, teams are expected to provide code for building the Client Container (see Section 5 for development resources). After the submission deadline, AWS administrators will pull submitted code and build the Client Container (see Section 6 for details on submission procedure). This Client Container will then run alongside the Evaluator within the JHU/APL network. During evaluation, the Evaluator sends input data to the Client Container and receives prediction responses via Rabbit MQ’s messaging protocol (see Section 3 for more information about the evaluation process and message formats). At the end of the evaluation, the system produces reports with Client responses and performance metrics for each team.

The Rabbit MQ Server and Evaluator will be developed and maintained by JHU/APL. Software submissions must comply with the JHU/APL evaluation system in order to be evaluated. JHU/APL will also provide a Client Shell as a starting point for submission preparation (see Section 5.2). To confirm compliance with the evaluation system, the JHU/APL team will provide a Continuous Integration/Continuous Development (CI/CD) system integrated with the DTC AWS network and teams’ code repositories that will build provided code and perform an abbreviated evaluation run to test submission compliance (see Section 5.4). Successful test outcome from the CI/CD indicates that the submission is compliant with the evaluation system. Teams will then need to follow the submission procedure to provide code for evaluation (see Section 6).



**Figure 1: High-level architecture of the JHU/APL evaluation system.** This system is composed of 1) the Evaluator, 2) Rabbit MQ server, and 3) the Client Container, containing the team’s model. The Evaluator sends test data to the Client Container using RabbitMQ messages, and the Client Container responds with the LSI predictions.

In Section 3.1, we describe the main responsibilities of the Evaluator and Client Container and the progression of messages passed between these two components during an evaluation run, as well as the results expected after evaluation. In Section 3.2, we describe the message formats expected between Evaluator and Client Container.

### 3.1 Evaluation Process

The Evaluator is responsible for accessing the held-out test dataset, packaging and serving data segments to the client container via RabbitMQ, and logging responses from the Client Container.

The Client Container is responsible for housing the submitted model, ingesting data provided by the Evaluator via RabbitMQ, aggregating or storing information across data segments (as needed), preprocessing data and running model inference, and sending model responses with LSI predictions back to the Evaluator.

After the Evaluator establishes connection with the Client Container, the Evaluator will begin sending data segments to the Client Container to perform model inferences and return LSI predictions. Evaluation occurs one case at a time with the Evaluator sending data segments within each case in temporal order from beginning to end.

The Evaluator and Client communicate using the following message formats (described in detail in Section 3.2):

- *Predict* message, containing a data segment provided to the Client for predicting LSIs.
- *Response* message, containing LSI predictions from the Client based off of data received
- *Acknowledge* message, containing status information about evaluation progress
- *Timed Out* message, indicating to the Client that the prediction process timed out
- *Error* message, indicating to the Client that the prediction response was mal-formed
- *Cleanup* message, indicating to the Client that the case is complete and evaluation is continuing with the next case

The following steps describe the sequence of messages sent between the Evaluator and the Client Container during an evaluation run:

- 1) **Evaluator establishes connection to the Client Container.** An initial message is sent from the Evaluator to the Client Container to establish the RabbitMQ channel for the evaluation run.
- 2) **Evaluator sends Predict message.** The Evaluator sends a new data segment in a Predict message and waits for a Response message from the Client.
- 3) **Client performs inference.** The Client receives the Predict message, performs any pre-processing activities and model inference.
- 4) **Client sends Response message.** The Client sends a response containing the list of predicted LSIs (if any).
- 5) **Evaluator waits for Response message and saves output.** The Evaluator saves the Response message from the Client for calculating metrics. If the Response message is not received from the Client before the segment duration has elapsed, go to Step 6; if the Response is timely but not well-formed, go to Step 7; otherwise, go to Step 8.
- 6) **(If applicable) Evaluator sends a Timed Out message and resumes with next case.** If the Client does not produce Response message within time limit (segment duration), then the Evaluator sends a Timed Out message to the Client. The evaluation is then interrupted and resumes at Step 1, starting with the next case in the test dataset.
- 7) **(If applicable) Evaluator sends an Error message and resumes with next case.** If the Client does not produce a well-formed Response message (e.g., LSI label is misspelled),

then the Evaluator sends an Error message to the Client. The evaluation is then interrupted and resumes at Step 1, starting with the next case in the test dataset.

- 8) **Evaluator sends Acknowledge message.** The Evaluator acknowledges receipt of a well-formed and timely Response message from the Client with status information about evaluation progress.
- 9) **Repeat Steps 2-8 for the remaining segments in the case.** The Evaluator continues sending incremental segments of the data for the current case. The Client Container is responsible for storing historical data from previous segments within the same case. For the last segment of the case, the Predict message will indicate the end-of-case using a boolean flag, indicating that the next Predict message will come from a new case.
- 10) **Evaluator confirms end of case with a Cleanup message.** After the prediction from the last segment of a case is received, the Evaluator sends a “clean-up” message to cue the Client to execute any process before beginning a new case.
- 11) **Repeat Steps 2-10 for remaining cases.** The Evaluator continues until all cases have been evaluated.

### 3.1.1 Response log

With each well-formed and timely response from the Client Container (Step 5 above), the following information will be stored to facilitate post-hoc metrics calculation and scoring:

- Patient case identifier
- Segment identifier
- Time window of evaluation segment
- Time elapsed from message sent to response received
- LSI predictions
- Optional response fields (e.g., embeddings, see Section 3.1.7)

This log will be provided to teams after each workshop and challenge event.

### 3.1.2 Metrics log

After an evaluation run, metrics will be computed from the Client response logs and the test dataset ground truth. This log will include the Mean Squared Correct by case, as described in the Rules Document.

## 3.2 Communication Protocol

In this section, we describe the contents and format of the following messages sent between the Evaluator and Client: *Predict*, *Acknowledge*, *Timed Out*, *Cleanup*, and *Error*. Further details on implementation are described in Section 5.

### 3.2.1 Predict Message Format

The Predict message is sent by the Evaluator to the Client for each data segment. It contains EHR data, VS data, and segment metadata. Here is an example of the Predict message format for a single segment:

```
{
```

```

"segment_id": "OUmJvhCVC5",
"case_id": "wv9ufeR87l",
"end_of_case": False,
"start_time": 900,
"stop_time": 1800,
"ehr": <dict>,
"vs": <dict>
}

```

### Message 1: Predict message format

Definitions for Predict message fields are as follows:

- *segment\_id* is a unique string identifier for the current segment
- *case\_id* is a unique string identifier for the current case, composed of many segments
- *end\_of\_case* is a boolean indicator that this is the last segment for the current case
- *start\_time* is the beginning of the segment as time elapsed from start of case in seconds
- *stop\_time* is the end of the segment as time elapsed from start of case in seconds
- *ehr* is a dictionary containing Electronic Health Record (EHR) data, described below
- *vs* is a dictionary containing Vital Signs (VS) data, described below

Depending on data availability within each case and segment, the *ehr* or *vs* field may be empty dictionaries or *None*. If present, all fields within these dictionaries are optional. For any available data, field names within both the *ehr* and *vs* dictionaries are consistent with the data dictionary and documentation provided with the training dataset. However, the structure of these dictionaries was simplified from the training dataset as described below.

In response to the Predict message, the Client is expected to respond with a Response message (see Section 3.2.2).

#### 3.2.1.1 *ehr Data Field*

The *ehr* dictionary contains EHR data made available according to the time bounds of the segment, which in total is only a subset of the full training dataset provided. Data are grouped into the following categories:

- **Start-of-Case.** Data available at the beginning of the case, such as GCS taken at the scene, injury type, and general demographic information.
- **At-Admission.** Data available at hospital admission, for example vitals taken at admission.
- **Event Time.** Timestamped data provided according to the segment time window in which they occur, along with timestamps relative to the beginning of the case (in seconds), for example procedures, labs, and medications.

Note that a portion of the EHR training dataset falls in none of these categories and will be excluded completely from the evaluation (for example, fields related to outcome or information not available in the acute period of treatment). [In addition, team solutions will be evaluated in multiple runs with further limited data sources available at test time.](#) EHR data has been divided into three categories to determine which data is available in which run: basic EHR, Expanded EHR, and LSI data. [Appendix A includes the list of possible fields provided during evaluation, the time](#)

when they are provided, and EHR data category for each field. See the [Data Competition Rules Document](#) for more information about evaluation runs.

### 3.2.1.2 vs Data Field

The vs dictionary contains Vital Signs (VS) data made available according to the time bounds of the segment. Timeseries data will have timestamps relative to the start of the case (in seconds). The vs dictionary will include any available trends and signal data from pre-hospital and in-hospital VS data with an internal structure that mirrors the file structure provided in the training dataset.

Here is an example of the vs dictionary containing all VS data sources:

```
{
  "in_hospital":
  {
    "signal": <dict>,
    "trends": <dict>
  },
  "pre_hospital":
  {
    "signal": <dict>,
    "trends": <dict>
  }
}
```

All signal and trends fields contain dictionaries that match the structure and field names of their respective HDF5 files in the training dataset and the accompanying dataset documentation. Only available data will be included, so all fields are optional.

### 3.2.2 Response Message Format

In response to the Predict message, Clients are expected to respond with LSI predictions [as a dictionary with key-value pairs with LSI label and confidence score](#). Confidence scores must be a value between 0 and 1, inclusive. The dictionary should contain a key for each of the possible LSI labels (see [Table 1](#)). Here is an example Response message:

```
{
  "segment_id": "OUmJvhCVC5",
  "lsi_predictions":
  {
    "airway_and_respiration": 0.2730309253546964,
    "bleeding_control": 0.06604488945823347,
    "blood_products": 0.00642003897813202,
    "chest_decompression": 0.18217806794637986,
    "neurologic_products_and_procedures": 0.7302976346657081,
    "damage_control_procedures": 0.28295521562222375
  }
}
```

**Message 2: Response message format**



Definitions of the Response message fields are as follows:

- *segment\_id* is the unique string identifier for the segment provided in the input message
- *lsi\_predictions* is a dictionary of key-value pairs with LSI label and confidence score (value between 0 and 1, inclusive). There should be a key for each LSI label.

Both fields above are required in the Response message. For specification of optional fields containing embeddings, see Section 3.2.7.

Table 1 contains the string labels for each LSI group expected in the *lsi\_predictions* dictionary keys, where LSI groups correspond to those used in *LSI\_table.csv* in the training dataset. **Note that LSI groups for Phase 2 have changed from Phase 1.**

**Table 1: LSI group response labels**

LSI GROUP	LABEL
Airway & Respiration	airway_and_respiration
Bleeding Control	bleeding_control
Blood Products	blood_products
Chest Decompression	chest_decompression
Neurologic Products & Procedures	neurologic_products_and_procedures
Damage Control Procedures	damage_control_procedures

After the Response message is received by the Evaluator, the Evaluator will send an Acknowledge message (see Section 3.2.3), and the evaluation will continue with the next segment. A Response message must be received within the segment duration from when the corresponding Predict message was sent, otherwise a Timed Out message is sent by the Evaluator (see Section 3.2.6).

### 3.2.3 Acknowledge Message Format

The Evaluator will send an Acknowledge message to the Client to indicate successful receipt of a prediction. Here is an example of an Acknowledge message:

```
{
  "case_id": "s1kojt25",
  "segment_id": "OUmJvhCVC5",
  "delta_runtime_sec": 0.2,
  "runtime_remaining_sec": 144000.8,
  "cases_remaining": 240
}
```

#### Message 3: Acknowledge message format

Definitions of the Acknowledge message fields are as follows:

- *case\_id* is the unique string identifier for the current case
- *segment\_id* is the unique string identifier for the current segment
- *delta\_runtime\_sec* is the time elapsed in seconds from Predict message sent to Response message received by the Evaluator, which contributes to the total evaluation runtime

Distribution Statement 'A' (Approved for Public Release, Distribution Unlimited)

- *runtime\_remaining\_sec* is the total evaluation runtime limit minus the cumulative runtime in seconds
- *cases\_remaining* is the number of cases remaining in the evaluation

There is no specific content required from the Client in response to an Acknowledge message.

### 3.2.4 Cleanup Message Format

Following the last segment in a case, the Evaluator will send a Cleanup message to the Client indicating the case has ended and the next Predict message will start a new case. Here is an example of a Cleanup message:

```
{
  "case_id": "s1kojt25",
  "runtime_remaining_sec": 144000.8,
  "cases_remaining": 240
}
```

#### Message 4: Cleanup message format

Definitions of the Cleanup message fields are as follows:

- *case\_id* is the unique string identifier for the current case
- *runtime\_remaining\_sec* is the total evaluation runtime limit minus the cumulative runtime in seconds
- *cases\_remaining* is the number of cases remaining in the evaluation

There is no specific content required from the Client in response to a Cleanup message.

### 3.2.5 Error Message Format

In the event of an error in the evaluation (e.g., malformed Response message from the Client), the Evaluator will send an Error message to the Client. Here is an example of an Error message:

```
{
  "case_id": "s1kojt25",
  "segment_id": "OUmJvhCVC5",
  "error_message": <string>
}
```

#### Message 5: Error message format

Definitions of the Error message fields are as follows:

- *case\_id* is the unique string identifier for the current case
- *segment\_id* is the unique string identifier for the current segment
- *error\_message* is a free-text message describing the error encountered

There is no specific content required from the Client in response to an Error message.

### 3.2.6 Timed Out Message Format

After a Predict message is sent, the corresponding Response message must be received from the Client before the time elapsed exceeds the segment duration. If no Response message is received before this time limit, a Timed Out message is sent by the Evaluator to the Client.

Here is an example of a Timed Out message sent by the Evaluator to the Client:

```
{
  "case_id": "s1kojt25",
  "segment_id": "OUmJvhCVC5",
  "delta_runtime_sec": 0.2,
  "runtime_remaining_sec": 144000.8,
  "cases_remaining": 240
}
```

**Message 6: Timed Out message format**

Definitions of the Timed Out message fields are as follows:

- *case\_id* is the unique string identifier for the current case
- *segment\_id* is the unique string identifier for the current segment
- *delta\_runtime\_sec* is the time elapsed in seconds from Predict message sent to Response message received by the Evaluator
- *runtime\_remaining\_sec* is the total evaluation runtime limit minus the cumulative runtime in seconds
- *cases\_remaining* is the number of cases remaining in the evaluation

There is no specific content required from the Client in response to a Timed Out message.

**3.2.7 Optional Response Message Fields**

To support greater interpretability of and ability to analyze performance of competitors’ models, there are additional fields that can be returned in the Response message. All of these fields are optional and do not factor into a competitor’s event score.

Many machine learning models operate by constructing vector representations (also called embeddings) of input data before performing additional processing to produce an output prediction. Providing access to this internal data representation can help improve interpretability of model behavior and provide additional surface area to diagnose and resolve issues with models. The schema for returning these embedding values is meant to help support this type of analysis. Given that there are multiple levels at which teams’ models may be operating, we suggest fields for embeddings representing both the segment level and the cumulative level, as well as a catch-all “other” category. None of this is prescriptive of how competitors approach the challenge, but rather a best hypothesis for the kinds of model-internal information that may be available.

Here is an example Response message that includes the optional embeddings field:

```
{
```

```
"segment_id": "OUmJvhCVC5",
"lsi_predictions": <dict>,
"embeddings": <dict>
}
```

### Message 7: Response message format with optional embeddings field

Optional fields are defined as follows and described in detail in subsections below:

- *embeddings* is a dictionary containing embeddings at different levels of internal data analysis

#### 3.2.7.1 *embeddings* Response Field

The *embeddings* response field contains a dictionary with the following format:

```
{
  "vs_segment": vs_segment_embedding,
  "vs_cumulative": vs_cumulative_embedding,
  "ehr_segment": ehr_segment_embedding,
  "ehr_cumulative": ehr_cumulative_embedding,
  "case_segment": case_segment_embedding,
  "case_cumulative": case_cumulative_embedding,
  "other": [other_embeddings_1, other_embeddings_2, ...]
}
```

### Message 8: *embeddings* dictionary format

Note that each individual field in the dictionary above is optional. The value for each embedding vector is expected to be a list of numbers, except for the “other” field, which is expected to be a list of such embedding vectors (i.e., a list of lists).

Embedding vector dimensionality may differ between fields; however, for a given field, the dimensionality should be consistent across all segments. For example, “vs\_segment” and “vs\_cumulative” embeddings may have different dimensionality, but their respective dimensionality should be the same for each data segment.

Embedding fields description:

*vs\_segment*

An embedding representing the vital signs data from the current data segment.

*vs\_cumulative*

An embedding representing the cumulative vital signs signals across prior segments up to and including this segment.

*ehr\_segment*

An embedding representing the EHR data from the current data segment.

*ehr\_cumulative*

An embedding representing the cumulative EHR data across prior segments up to and including this segment.

*case\_segment*

A higher-level embedding representing the overall case status given the current segment.

*case\_cumulative*

Distribution Statement ‘A’ (Approved for Public Release, Distribution Unlimited)

A higher-level embedding representing the overall case status given the cumulative data across prior segments up to and including this segment.

*other*

Any other embedding(s) that do not fit into the schema above may be included here.

### 3.2.7.2 Response Message Format, Including Optional Fields

Given the above specification for the optional fields “embeddings” and “lsi\_confidence\_scores”, here is an example client response message that includes the optional fields:

```
{
  "segment_id": "OUMjvhCVC5",
  "lsi_predictions": {
    "airway_and_respiration": 0.0015,
    "bleeding_control": 0.01163,
    ...
  },
  "embeddings": {
    "case_segment": [0.152, 0.023, ..., 0.134],
    "case_cumulative": [0.693, 0.193, ..., 0.081]
  },
  "other": [
    [0.232, 0.024, ..., 0.817],
    [0.251, 0.004, ..., 0.948]
  ]
}
```

**Message 10: Example response including optional diagnostic fields**

## 4 Evaluation Environment

We are expecting the following hardware environment and constraints for the Phase 2 Challenge Event:

CPU	1 CPU w/ 8 cores @ 3.50 GHz
RAM	100 GB
GPU	Titan V (12 GB VRAM, single GPU)
GPU DRIVER	NVIDIA-DRIVER Version 510.47.03
NETWORK	No access
TEST CASES	TBD
TIME LIMITATION	48 hours

## 5 Submission Development

Each team must provide code that is compliant with the evaluation process. The following section describes submission requirements, resources, and tools for testing with the JHU/APL evaluation system.

### 5.1 Requirements

The following subsections describe the minimum requirements for successful submissions.

#### 5.1.1 DTC Base Image

Submissions must be built from an approved DTC Base Image containing the required package *dtc\_messaging* for interacting with the Evaluator. The following base images are provided in AWS Elastic Container Registry (ECR):

- *dtc-base-image:latest* for GPU support in Client development
- *dtc-base-image-cpu:latest* for CPU-only support in Client development

Code used to build these images is provided on AWS CodeCommit: <https://git-codecommit.us-east-1.amazonaws.com/v1/repos/dtc-base-image>

#### 5.1.2 DTC Base Model

Submissions must use a model class that inherits from *DTC\_BaseModel* class (included in the *dtc\_messaging* package within the DTC Base Image). Callback functions are used to respond to message types described in Section 3. The model class must implement the following callback methods:

- `predict()`: receives a Predict message and returns a Response message
- `acknowledge()`: receives an Acknowledge message, no return message required
- `error()`: receives an Error message, no return message required
- `timed_out()`: receives a Timed Out message, no return message required
- `cleanup()`: receives a Cleanup message, no return message required

An example implementation of the *DTC\_BaseModel* class is provided in `template_model.py` in the Client Shell (see Section 5.2).

#### 5.1.3 Docker Entrypoint

The following entrypoint and command will be used to run Client Container submissions within the evaluation system:

```
command: "./run_client.py --host <host> --queue <queue>"
entrypoint: ["python3"]
```

where `<host>` and `<queue>` will be modified at evaluation time to set the host and RabbitMQ queue name. Submissions are required to provide `run_client.py` in the working directory of

the Docker container with command line arguments `--host` and `--queue`. Any additional command line arguments will be ignored during evaluation. Additional parameters required by the submission may be included in a configuration file within the Docker container.

An example implementation of the `run_client.py` script is provided in the Client Shell (see Section 5.2).

#### 5.1.4 Client Log

A volume will be mounted to the Client Container where log files may be saved for delivery back to teams in AWS after an evaluation event. This is an optional feature, and any log files will not impact team scores, nor will they be shared between teams. Any log files should be given unique filenames (e.g., with timestamp) so as not to be overwritten by subsequent evaluation runs. Importantly, EHR or VS data (including derived data) should not be saved in log files.

The following directory will be mounted as a volume to the Client Container for saving log files within the Docker container: `/usr/src/app/logs`

## 5.2 Client Shell

To assist in developing compliant submissions, JHU/APL has provided a Client Shell that includes the minimal code needed to create a functioning Client Container. Teams may incorporate their own packages and model-specific code to the Client Shell to build their submission.

The Client Shell contains the following resources:

- `run_client.py`: script used to run the Client
- `template_model.py`: model implementation of *DTC\_BaseModel*
- `Dockerfile`: example to build Client Container from *dtc-base-image* and install additional dependencies
- `ReadMe.md`: additional information on Client Shell usage, also available on the AWS wiki at: [https://jhuapl-dtc-ta2.github.io/wiki/running\\_client\\_shell/](https://jhuapl-dtc-ta2.github.io/wiki/running_client_shell/)

The Client Shell is provided on AWS CodeCommit: <https://git-codecommit.us-east-1.amazonaws.com/v1/repos/client-shell>

## 5.3 Evaluator Container Testing

Docker images containing the DTC Evaluator (*dtc-evaluator:latest*) and the RabbitMQ Server (*dtc-rabbitmq:latest*) will be released to participants to assist with testing Clients within AWS prior to submitting for evaluation. See future posts to the DTC forum and documentation on the AWS wiki for more information.

## 5.4 CodeBuild (CI/CD) Compliance Testing

JHU/APL will provide a Continuous Integration/Continuous Development (CI/CD) system within AWS to give automatic feedback on submission compliance with the evaluation system. This system will containerize selected code using a standard *buildspec* and perform an abbreviated evaluation run to assess the code's compliance with the evaluation system.

To submit code to the CI/CD system, teams should push code to the *compliance-test* branch of their team repository. This action will automatically trigger the CI/CD system to pull the repository and build code with the most recent commit on the *compliance-test* branch. Once the build is complete, the CI/CD system will perform an evaluation run using a small validation dataset to assess compliance with the evaluation system. Any artifacts (e.g., logs) produced by the compliance test will be provided to teams for review within their scratch bucket under the *build\_logs* directory.

Usage of the CI/CD system is voluntary, and there is no limit to the number of times a team can test code through the CI/CD system. However, costs related to running the CI/CD system will be subtracted from the team's budget. These costs are expected to be minimal, but code should be pushed to the *compliance-test* branch sparingly to minimize budget usage.

Additional information and release updates to the CI/CD system will be posted to the DTC Forum.

## 6 Submission Procedure

Code will be pulled down from CodeCommit upon the submission deadline using a specific git tag pushed to the team repository on CodeCommit. Tag nomenclature for event submissions should use the following convention:

*submission-phase<PHASE\_NUMBER>-<EVENT\_TYPE>*

where  $PHASE\_NUMBER \in \{1, 2, 3\}$  and  $EVENT\_TYPE \in \{\text{workshop, challenge}\}$ . For example, for the Phase 2 workshop event, submissions should be tagged as:

*submission-phase2-workshop*

This procedure ensures that a specific, unambiguous commit is evaluated as the official submission. The git tag may be moved to a different commit ahead of the submission deadline, however changing the git tag after submission deadline will not be possible. The submission git tag will also trigger a CI/CD compliance test to ensure submitted code complies with the evaluation system.



## 7 Appendix A – Fields Provided During Evaluation

The following tables provide information about which data fields from UMB and UPitt datasets are provided during evaluation.

In each table, the *Provided At* column specifies when the field is provided during the case: “start” fields are provided at start of case, “admission” fields are provided at hospital admission”, and “timestamp” fields are provided at the record timestamp (“elapsed\_from\_start” within the same table). The *Type* column groups data into “Basic EHR”, “Expanded EHR”, and “LSI”. The *Type* determines which run(s) each field is provided. See the Data Competition Rules Document for more information about evaluation runs.

**Table A.1: UMB fields**

Table	Field	Provided At	Type
demo_scores	ADM_DYSBP	admission	Basic EHR
demo_scores	ADM_GCS_EYE	admission	Basic EHR
demo_scores	ADM_GCS_MOTOR	admission	Basic EHR
demo_scores	ADM_GCS_VERBAL	admission	Basic EHR
demo_scores	ADM_HR	admission	Basic EHR
demo_scores	ADM_O2SAT	admission	Basic EHR
demo_scores	ADM_RR	admission	Basic EHR
demo_scores	ADM_SYSBP	admission	Basic EHR
demo_scores	ADM_TEMP	admission	Basic EHR
demo_scores	elapsed_from_start	admission	Basic EHR
demo_scores	GCSTOTAL	admission	Basic EHR
demo_scores	RTS_A	admission	Basic EHR
demo_scores	AGE	start	Basic EHR
demo_scores	HEIGHT	start	Basic EHR
demo_scores	INJTYPEID	start	Basic EHR
demo_scores	INJURYTYPEDESCRIP	start	Basic EHR
demo_scores	race_descrip	start	Basic EHR
demo_scores	RACEID	start	Basic EHR
demo_scores	RTS_S	start	Basic EHR
demo_scores	SEXID	start	Basic EHR
demo_scores	WEIGHT	start	Basic EHR
ems	LANDVSAIR	start	Expanded EHR
ems_addition	DESCRIP	timestamp	Expanded EHR
ems_addition	elapsed_from_start	timestamp	Expanded EHR
ems_addition	STARTTIME	timestamp	Expanded EHR
labs	COMPTEXT	timestamp	Expanded EHR

labs	elapsed_from_start	timestamp	Expanded EHR
labs	lnc_code	timestamp	Expanded EHR
labs	OBSDATETIME_elapsed_from_start	timestamp	Expanded EHR
labs	result_time_elapsed_from_start	timestamp	Expanded EHR
labs	RSLT	timestamp	Expanded EHR
labs	TESTTEXT	timestamp	Expanded EHR
labs	UNITS	timestamp	Expanded EHR
LSI_table	elapsed_from_start	timestamp	LSI
LSI_table	in_hospital	timestamp	LSI
LSI_table	lsi_description	timestamp	LSI
LSI_table	lsi_group	timestamp	LSI
medications	display_name	timestamp	Expanded EHR
medications	dose_unit	timestamp	Expanded EHR
medications	elapsed_from_start	timestamp	Expanded EHR
medications	mar_action	timestamp	Expanded EHR
medications	pat_loc	timestamp	Expanded EHR
medications	pharm_class	timestamp	Expanded EHR
medications	pharm_subclass	timestamp	Expanded EHR
medications	route	timestamp	Expanded EHR
medications	sig	timestamp	Expanded EHR
medications	thera_class	timestamp	Expanded EHR
non_op_procs	DESCRIP	timestamp	Expanded EHR
non_op_procs	elapsed_from_start	timestamp	Expanded EHR
operations	CPT	timestamp	Expanded EHR
operations	OR_START_TIME_elapsed_from_start	timestamp	Expanded EHR
operations	OR_STOP_TIME_elapsed_from_start	timestamp	Expanded EHR
operations	PROCTEXT	timestamp	Expanded EHR
operations	SERVICE_DESCRIP	timestamp	Expanded EHR
operations	SERVICEID	timestamp	Expanded EHR
pta_vitals	elapsed_from_start	timestamp	Basic EHR
pta_vitals	PTA_DBP	timestamp	Basic EHR
pta_vitals	PTA_GCS_E	timestamp	Basic EHR
pta_vitals	PTA_GCS_M	timestamp	Basic EHR
pta_vitals	PTA_GCS_TOTAL	timestamp	Basic EHR
pta_vitals	PTA_GCS_V	timestamp	Basic EHR
pta_vitals	PTA_HR	timestamp	Basic EHR
pta_vitals	PTA_RR	timestamp	Basic EHR
pta_vitals	PTA_SBP	timestamp	Basic EHR
pta_vitals	PTA_TEMP	timestamp	Basic EHR

**Table A.2: UPitt fields**

Table	Field	Provided At	Type
airway	AirwayOutcome	admission	Expanded EHR
airway	AirwayPerformedBy	admission	Expanded EHR
airway	AirwaySecuredVia	admission	Expanded EHR
airway	AirwayStatus	admission	Expanded EHR
airway	AirwayVerification	admission	Expanded EHR
airway	AirwayVerification1	admission	Expanded EHR
airway	AirwayVerification2	admission	Expanded EHR
airway	AirwayVerification3	admission	Expanded EHR
airway	AirwayVerification4	admission	Expanded EHR
airway	AirwayVerification5	admission	Expanded EHR
airway	AirwayVerification6	admission	Expanded EHR
burns	BurnsMethod	start	Expanded EHR
burns	BurnsPerc1stDegree	start	Expanded EHR
burns	BurnsPerc2ndDegree	start	Expanded EHR
burns	BurnsPerc3rdDegree	start	Expanded EHR
burns	BurnsPercArmsAnterior	start	Expanded EHR
burns	BurnsPercArmsPoterior	start	Expanded EHR
burns	BurnsPercHeadAnterior	start	Expanded EHR
burns	BurnsPercHeadPoterior	start	Expanded EHR
burns	BurnsPercLegsAnterior	start	Expanded EHR
burns	BurnsPercLegsPoterior	start	Expanded EHR
burns	BurnsPercTrunkAnterior	start	Expanded EHR
burns	BurnsPercTrunkPoterior	start	Expanded EHR
burns	CarboninMouth	start	Expanded EHR
burns	CO	start	Expanded EHR
burns	PoorVentilation	start	Expanded EHR
burns	SingedNasalHairs	start	Expanded EHR
burns	TBSA	start	Expanded EHR
burns	Type	start	Expanded EHR
cardiac	AEDUse	admission	Expanded EHR
cardiac	CardiacArrestPresent	admission	Expanded EHR
cardiac	CPRBy	admission	Expanded EHR
cardiac	CPRTType	admission	Expanded EHR
cardiac	CPRTType1	admission	Expanded EHR
cardiac	CPRTType2	admission	Expanded EHR
cardiac	CPRTType3	admission	Expanded EHR
cardiac	CPRTType4	admission	Expanded EHR

cardiac	DefibBy	admission	Expanded EHR
cardiac	DefibrillatorType	admission	Expanded EHR
cardiac	DestRhythm	admission	Expanded EHR
cardiac	EndEvent	admission	Expanded EHR
cardiac	Etiology	admission	Expanded EHR
cardiac	InitialRhythm	admission	Expanded EHR
cardiac	ReasonTerminated	admission	Expanded EHR
cardiac	Resuscitation1	admission	Expanded EHR
cardiac	Resuscitation2	admission	Expanded EHR
cardiac	TherapeuticHypothermia	admission	Expanded EHR
cardiac	WhoWitnessed	admission	Expanded EHR
cardiovascular	CapillaryRefill	admission	Expanded EHR
cardiovascular	Edema	admission	Expanded EHR
cardiovascular	InvasiveArterialLine	admission	Expanded EHR
cardiovascular	JVD	admission	Expanded EHR
cardiovascular	PulseBrachialL	admission	Expanded EHR
cardiovascular	PulseBrachialR	admission	Expanded EHR
cardiovascular	PulseCarotidL	admission	Expanded EHR
cardiovascular	PulseCarotidR	admission	Expanded EHR
cardiovascular	PulseFemoralL	admission	Expanded EHR
cardiovascular	PulseFemoralR	admission	Expanded EHR
cardiovascular	PulseRadialL	admission	Expanded EHR
cardiovascular	PulseRadialR	admission	Expanded EHR
cardiovascular	TempDegreesType	admission	Expanded EHR
cardiovascular	Temperature	admission	Expanded EHR
cardiovascular	TemperatureObtainedMethod	admission	Expanded EHR
drain	ChestTubeSizeL	admission	Expanded EHR
drain	ChestTubeSizeR	admission	Expanded EHR
drain	ChestTubeSuctionL	admission	Expanded EHR
drain	ChestTubeSuctionR	admission	Expanded EHR
drain	Colostomy	admission	Expanded EHR
drain	FoleySize	admission	Expanded EHR
drain	Hematuria	admission	Expanded EHR
drain	Ileostomy	admission	Expanded EHR
drain	NGSuction	admission	Expanded EHR
drain	NGTubeSize	admission	Expanded EHR
drain	OGSuction	admission	Expanded EHR
drain	OGTube	admission	Expanded EHR
drain	Other	admission	Expanded EHR

drain	PEGCapped	admission	Expanded EHR
drain	PEGTube	admission	Expanded EHR
drain	Ventriculostomy	admission	Expanded EHR
drain	VentriculostomyColor	admission	Expanded EHR
events	BloodPressureDiastolic	timestamp	Basic EHR
events	BloodPressureSystolic	timestamp	Basic EHR
events	BPMMethod	timestamp	Basic EHR
events	Carboxyhemoglobin	timestamp	Basic EHR
events	ECGMethod	timestamp	Basic EHR
events	elapsed_from_start	timestamp	Basic EHR
events	EndotrachealCO2	timestamp	Basic EHR
events	EndotrachealCO2Type	timestamp	Basic EHR
events	GCSEye	timestamp	Basic EHR
events	GCSMotor	timestamp	Basic EHR
events	GCSVerbal	timestamp	Basic EHR
events	Glucose	timestamp	Basic EHR
events	HeartRate	timestamp	Basic EHR
events	HeartRateMethod	timestamp	Basic EHR
events	LevelofConsciousness	timestamp	Basic EHR
events	MeanArterialPressure	timestamp	Basic EHR
events	OxygenSaturation	timestamp	Basic EHR
events	Procedure	timestamp	Basic EHR
events	PulseRhythm	timestamp	Basic EHR
events	Respiration	timestamp	Basic EHR
events	RespiratoryEffort	timestamp	Basic EHR
events	RhythmCoded	timestamp	Basic EHR
events	TempDegreesType	timestamp	Basic EHR
events	Temperature	timestamp	Basic EHR
events	TemperatureObtainedMethod	timestamp	Basic EHR
events	VitalRhythms	timestamp	Basic EHR
fluids	INBeforeCOLL	start	Expanded EHR
fluids	INBeforeCrys	start	Expanded EHR
fluids	INBeforeOther	start	Expanded EHR
fluids	OUTBeforeEBL	start	Expanded EHR
fluids	OUTBeforeOther	start	Expanded EHR
fluids	OUTBeforeUO	start	Expanded EHR
hosp_trqt	trqt	admission	Expanded EHR
hosp_trqt	trqt_place	admission	Expanded EHR
hosp_trqt	trqt_type	admission	Expanded EHR

hosp_trqt	trqt_use	admission	Expanded EHR
hosp_wide	age	admission	Basic EHR
hosp_wide	ethnicity	admission	Basic EHR
hosp_wide	etiology	admission	Basic EHR
hosp_wide	gcs	admission	Basic EHR
hosp_wide	gcs_m	admission	Basic EHR
hosp_wide	inj_type	admission	Basic EHR
hosp_wide	lsys_bp_1	admission	Basic EHR
hosp_wide	lsys_bp_2	admission	Basic EHR
hosp_wide	lsys_bp_r	admission	Basic EHR
hosp_wide	lsys_bp_s	admission	Basic EHR
hosp_wide	par_drg_a	admission	Basic EHR
hosp_wide	pulse_a	admission	Basic EHR
hosp_wide	pupil_a	admission	Basic EHR
hosp_wide	race	admission	Basic EHR
hosp_wide	resp	admission	Basic EHR
hosp_wide	rts	admission	Basic EHR
hosp_wide	sao2_a	admission	Basic EHR
hosp_wide	sbp	admission	Basic EHR
hosp_wide	sedated_a	admission	Basic EHR
hosp_wide	sex	admission	Basic EHR
hosp_wide	sign_life	admission	Basic EHR
hosp_wide	temp_c	admission	Basic EHR
hosp_wide	base_def	admission	Expanded EHR
hosp_wide	blood_pre	admission	Expanded EHR
hosp_wide	destination	admission	Expanded EHR
hosp_wide	extric_s	admission	Expanded EHR
hosp_wide	hgt_fall	admission	Expanded EHR
hosp_wide	hospital	admission	Expanded EHR
hosp_wide	intubat_a	admission	Expanded EHR
hosp_wide	transfer	admission	Expanded EHR
injurydetails	InjuryCause	start	Basic EHR
injurydetails	InjuryMechanism	start	Basic EHR
injurydetails	InjuryMechanism1	start	Basic EHR
injurydetails	InjuryMechanism2	start	Basic EHR
injurydetails	InjuryMechanism3	start	Basic EHR
injurydetails	DrugsAlcohol	start	Expanded EHR
injurydetails	DrugsAlcoholIndicator	start	Expanded EHR
injurydetails	DrugsAlcoholIndicator1	start	Expanded EHR

injurydetails	DrugsAlcoholIndicator2	start	Expanded EHR
injurydetails	DrugsAlcoholIndicator3	start	Expanded EHR
injurydetails	DrugsAlcoholIndicator4	start	Expanded EHR
injurydetails	DrugsAlcoholIndicator5	start	Expanded EHR
injurydetails	FallHeight	start	Expanded EHR
injurydetails	FallHeightUnits	start	Expanded EHR
injurydetails	FallSurface	start	Expanded EHR
injurydetails	IntentionalInjury	start	Expanded EHR
injurydetails	LandedOn	start	Expanded EHR
injurydetails	ReasonForEncounter	start	Expanded EHR
injurydetails	TraumaCenterCriterion	start	Expanded EHR
injurydetails	TraumaCenterCriterion1	start	Expanded EHR
injurydetails	TraumaCenterCriterion2	start	Expanded EHR
injurydetails	TraumaCenterCriterion3	start	Expanded EHR
injurydetails	TraumaCenterCriterion4	start	Expanded EHR
injurydetails	TraumaCenterCriterion5	start	Expanded EHR
injurydetails	WorkRelated	start	Expanded EHR
ipfgen	AssessmentLocation	timestamp	Expanded EHR
ipfgen	elapsed_from_start	timestamp	Expanded EHR
ipfmisc	AbdomenAppearance	start	Expanded EHR
ipfmisc	AbdomenPalpation	start	Expanded EHR
ipfmisc	BackBoardPTA	start	Expanded EHR
ipfmisc	BurnArea	start	Expanded EHR
ipfmisc	BurnBodySide	start	Expanded EHR
ipfmisc	BurnDegree	start	Expanded EHR
ipfmisc	BurnPercentage	start	Expanded EHR
ipfmisc	CCollarPTA	start	Expanded EHR
ipfmisc	CIDPTA	start	Expanded EHR
ipfmisc	ImmobilizationPTAOutcome	start	Expanded EHR
ipfmisc	ImmobilizationPTAPerformedBy	start	Expanded EHR
ipfmisc	Impression	start	Expanded EHR
ipfmisc	InjurySymptom	start	Expanded EHR
ipfmisc	KEDPTA	start	Expanded EHR
ipfmisc	PrimaryInjurySystem	start	Expanded EHR
ipfmisc	Trachea	start	Expanded EHR
ipfspc	MentalAssessment	start	Expanded EHR
ipfspc	NeurologicalAssessment	start	Expanded EHR
ipfspc	SkinAssessment	start	Expanded EHR
labs	ALB	timestamp	Expanded EHR

labs	AlcoholOnBreath	timestamp	Expanded EHR
labs	ALT	timestamp	Expanded EHR
labs	APAP	timestamp	Expanded EHR
labs	APHI	timestamp	Expanded EHR
labs	AST	timestamp	Expanded EHR
labs	BE	timestamp	Expanded EHR
labs	BGAccess	timestamp	Expanded EHR
labs	BILI	timestamp	Expanded EHR
labs	BloodAlcohol	timestamp	Expanded EHR
labs	BNP	timestamp	Expanded EHR
labs	BUN	timestamp	Expanded EHR
labs	CA	timestamp	Expanded EHR
labs	CL	timestamp	Expanded EHR
labs	CO2	timestamp	Expanded EHR
labs	COHB	timestamp	Expanded EHR
labs	CPK	timestamp	Expanded EHR
labs	CR	timestamp	Expanded EHR
labs	DDimer	timestamp	Expanded EHR
labs	elapsed_from_start	timestamp	Expanded EHR
labs	GLU	timestamp	Expanded EHR
labs	HcgSerum	timestamp	Expanded EHR
labs	HcgUrine	timestamp	Expanded EHR
labs	HCO3	timestamp	Expanded EHR
labs	HCT	timestamp	Expanded EHR
labs	HGB	timestamp	Expanded EHR
labs	INR	timestamp	Expanded EHR
labs	IonizedCalcium	timestamp	Expanded EHR
labs	K	timestamp	Expanded EHR
labs	LactateArterial	timestamp	Expanded EHR
labs	LactateVenous	timestamp	Expanded EHR
labs	Lipase	timestamp	Expanded EHR
labs	MG	timestamp	Expanded EHR
labs	NA	timestamp	Expanded EHR
labs	PCO2	timestamp	Expanded EHR
labs	PH	timestamp	Expanded EHR
labs	PLTS	timestamp	Expanded EHR
labs	PO2	timestamp	Expanded EHR
labs	PT	timestamp	Expanded EHR
labs	PTA	timestamp	Expanded EHR



labs	PTT	timestamp	Expanded EHR
labs	RBC	timestamp	Expanded EHR
labs	SAT	timestamp	Expanded EHR
labs	SVO2	timestamp	Expanded EHR
labs	TBILI	timestamp	Expanded EHR
labs	Troponin	timestamp	Expanded EHR
labs	WBC	timestamp	Expanded EHR
LSI_table	elapsed_from_start	timestamp	LSI
LSI_table	in_hospital	timestamp	LSI
LSI_table	lsi_description	timestamp	LSI
LSI_table	lsi_group	timestamp	LSI
neuro	ChemParalyzed	start	Basic EHR
neuro	InitialGCSEye	start	Basic EHR
neuro	InitialGCSMotor	start	Basic EHR
neuro	InitialGCSTotal	start	Basic EHR
neuro	InitialGCSVerbal	start	Basic EHR
neuro	RevisedTraumaScore	start	Basic EHR
neuro	RevisedTraumaScoreBP	start	Basic EHR
neuro	RevisedTraumaScoreResp	start	Basic EHR
neuro	LevelofConsciousness	start	Expanded EHR
neuro	LossofConsciousness	start	Expanded EHR
neuro	MotorLA	start	Expanded EHR
neuro	MotorLL	start	Expanded EHR
neuro	MotorRA	start	Expanded EHR
neuro	MotorRL	start	Expanded EHR
neuro	PupilReactivityL	start	Expanded EHR
neuro	PupilReactivityR	start	Expanded EHR
neuro	PupilSizeL	start	Expanded EHR
neuro	PupilSizeR	start	Expanded EHR
neuro	SensoryLA	start	Expanded EHR
neuro	SensoryLL	start	Expanded EHR
neuro	SensoryRA	start	Expanded EHR
neuro	SensoryRL	start	Expanded EHR
patient	AgeInYears	start	Basic EHR
patient	AgeType	start	Basic EHR
patient	BloodType	start	Basic EHR
patient	Gender	start	Basic EHR
patient	Height	start	Basic EHR
patient	HeightType	start	Basic EHR

patient	PatientAge	start	Basic EHR
patient	Race	start	Basic EHR
patient	Weight	start	Basic EHR
patient	WeightType	start	Basic EHR
patient	BarriersToCareNew	admission	Expanded EHR
patient	ChiefComplaint	start	Expanded EHR
patient	ChiefComplaintDuration	start	Expanded EHR
patient	MedicalNecessity	start	Expanded EHR
patient	PatientActivity	start	Expanded EHR
patient	ReasonforInterfacilityTransfer	start	Expanded EHR
priorivs	IVGauge	start	Expanded EHR
priorivs	IVRate	start	Expanded EHR
priorivs	IVSeq	start	Expanded EHR
priorivs	IVSite	start	Expanded EHR
priorivs	IVSolution	start	Expanded EHR
priorimeds	MedConcentration	start	Expanded EHR
priorimeds	MedDose	start	Expanded EHR
priorimeds	MedDrip	start	Expanded EHR
priorimeds	MedName	start	Expanded EHR
priorimeds	MedNameCoded	start	Expanded EHR
priorimeds	MedRoute	start	Expanded EHR
priorimeds	MedRouteName	start	Expanded EHR
priorimeds	MedSeq	start	Expanded EHR
priorimeds	RouteCoded	start	Expanded EHR
proclabs	elapsed_from_start	timestamp	Basic EHR
proclabs	LabLactateArterial	timestamp	Basic EHR
proclabs	LabLactateVenous	timestamp	Basic EHR
proclabs	LabsHCO3	timestamp	Basic EHR
proclabs	LabsPCO2	timestamp	Basic EHR
proclabs	LabspH	timestamp	Basic EHR
proclabs	LabspO2	timestamp	Basic EHR
proclabs	LabIonizedCalcium	timestamp	Expanded EHR
proclabs	LabPTA	timestamp	Expanded EHR
proclabs	LabsBE	timestamp	Expanded EHR
proclabs	LabsBGAccess	timestamp	Expanded EHR
proclabs	LabsCl	timestamp	Expanded EHR
proclabs	LabsGlu	timestamp	Expanded EHR
proclabs	LabsHgb	timestamp	Expanded EHR
proclabs	LabsK	timestamp	Expanded EHR

proclabs	LabsNa	timestamp	Expanded EHR
proclabs	LabsOther	timestamp	Expanded EHR
proclabs	LabsSAT	timestamp	Expanded EHR
proclabs	LabSVO2	timestamp	Expanded EHR
procmain	AirwayAction	timestamp	Expanded EHR
procmain	AirwayOxygenFlow	timestamp	Expanded EHR
procmain	CardiacAction	timestamp	Expanded EHR
procmain	CardiacDefibType	timestamp	Expanded EHR
procmain	CardiacPacingEnergy	timestamp	Expanded EHR
procmain	CardiacPacingMode	timestamp	Expanded EHR
procmain	CardiacPacingRate	timestamp	Expanded EHR
procmain	CardiacShockEnergy	timestamp	Expanded EHR
procmain	CardiacShockType	timestamp	Expanded EHR
procmain	ComplicationAirwayIntubation	timestamp	Expanded EHR
procmain	ComplicationAirwayIntubation1	timestamp	Expanded EHR
procmain	ComplicationAirwayIntubation2	timestamp	Expanded EHR
procmain	ComplicationAirwayIntubation3	timestamp	Expanded EHR
procmain	ComplicationAirwayIntubation4	timestamp	Expanded EHR
procmain	ComplicationAirwayIntubation5	timestamp	Expanded EHR
procmain	DrainAction	timestamp	Expanded EHR
procmain	DrainSide	timestamp	Expanded EHR
procmain	DrainSize	timestamp	Expanded EHR
procmain	DrainSizeUnit	timestamp	Expanded EHR
procmain	DrainSuction	timestamp	Expanded EHR
procmain	elapsed_from_start	timestamp	Expanded EHR
procmain	HospNotifyAlertType	timestamp	Expanded EHR
procmain	HospNotifyMethod	timestamp	Expanded EHR
procmain	HospNotifyRegistryCandidate	timestamp	Expanded EHR
procmain	ImmobOption	timestamp	Expanded EHR
procmain	InitiateIVGauge	timestamp	Expanded EHR
procmain	InitiateIVSite	timestamp	Expanded EHR
procmain	InitiateIVType	timestamp	Expanded EHR
procmain	IntubationCMatLips	timestamp	Expanded EHR
procmain	IntubationCuffFill	timestamp	Expanded EHR
procmain	IntubationCuffFillQty	timestamp	Expanded EHR
procmain	IntubationLaryngoscopicGrade	timestamp	Expanded EHR
procmain	IntubationLaryngoscopeBlade	timestamp	Expanded EHR
procmain	IntubationMallampati	timestamp	Expanded EHR
procmain	IntubationMethod	timestamp	Expanded EHR

procmain	IntubationRSIProtocol	timestamp	Expanded EHR
procmain	IntubationSellickManeuver	timestamp	Expanded EHR
procmain	IntubationSize	timestamp	Expanded EHR
procmain	IntubationStyletUsed	timestamp	Expanded EHR
procmain	MedicalConsultMethod	timestamp	Expanded EHR
procmain	MedicationDosage	timestamp	Expanded EHR
procmain	MedicationDosageUnit	timestamp	Expanded EHR
procmain	MedicationName	timestamp	Expanded EHR
procmain	MedicationRate	timestamp	Expanded EHR
procmain	MedicationRateUnit	timestamp	Expanded EHR
procmain	MedicationRoute	timestamp	Expanded EHR
procmain	Procedure	timestamp	Expanded EHR
procmain	ProcedureAttempts	timestamp	Expanded EHR
procmain	ProcedureAuthorization	timestamp	Expanded EHR
procmain	ProcedureComplication	timestamp	Expanded EHR
procmain	ProcedureComplication1	timestamp	Expanded EHR
procmain	ProcedureComplication2	timestamp	Expanded EHR
procmain	ProcedureComplication3	timestamp	Expanded EHR
procmain	ProcedureComplication4	timestamp	Expanded EHR
procmain	ProcedureComplication5	timestamp	Expanded EHR
procmain	ProcedureFailureReason	timestamp	Expanded EHR
procmain	ProcedureFailureReason1	timestamp	Expanded EHR
procmain	ProcedureFailureReason2	timestamp	Expanded EHR
procmain	ProcedureFailureReason3	timestamp	Expanded EHR
procmain	ProcedureFailureReason4	timestamp	Expanded EHR
procmain	ProcedureResponse	timestamp	Expanded EHR
procmain	ProcedureSuccessful	timestamp	Expanded EHR
procmain	TitrateMedication	timestamp	Expanded EHR
procmain	TitrateNewDose	timestamp	Expanded EHR
procmain	TitrateNewDoseRoute	timestamp	Expanded EHR
procmain	TitrateNewDoseUnit	timestamp	Expanded EHR
procother	elapsed_from_start	timestamp	Expanded EHR
procother	ImmobAssessment	timestamp	Expanded EHR
procother	ImmobAssessment1	timestamp	Expanded EHR
procother	ImmobAssessment2	timestamp	Expanded EHR
procother	ImmobAssessment3	timestamp	Expanded EHR
procother	ImmobAssessment4	timestamp	Expanded EHR
procother	ImmobAssessment5	timestamp	Expanded EHR
procother	ImmobilizationAction	timestamp	Expanded EHR

procother	ImmobilizationAction1	timestamp	Expanded EHR
procother	ImmobilizationAction2	timestamp	Expanded EHR
procother	ImmobilizationAction3	timestamp	Expanded EHR
procother	ImmobilizationAction4	timestamp	Expanded EHR
procother	ImmobilSite	timestamp	Expanded EHR
procother	ImmobilSite1	timestamp	Expanded EHR
procother	ImmobilSite2	timestamp	Expanded EHR
procother	ImmobilSite3	timestamp	Expanded EHR
procother	ImmobilSite4	timestamp	Expanded EHR
procother	IntubationIndication	timestamp	Expanded EHR
procother	IntubationIndication1	timestamp	Expanded EHR
procother	IntubationIndication2	timestamp	Expanded EHR
procother	IntubationIndication3	timestamp	Expanded EHR
procother	IntubationIndication4	timestamp	Expanded EHR
procother	IntubationIndication5	timestamp	Expanded EHR
procother	IntubationVerification	timestamp	Expanded EHR
procother	IntubationVerification1	timestamp	Expanded EHR
procother	IntubationVerification2	timestamp	Expanded EHR
procother	IntubationVerification3	timestamp	Expanded EHR
procother	IntubationVerification4	timestamp	Expanded EHR
procother	IntubationVerification5	timestamp	Expanded EHR
procother	IntubationVerification6	timestamp	Expanded EHR
procvent	AutoPeep	timestamp	Expanded EHR
procvent	BreathType	timestamp	Expanded EHR
procvent	BreathType1	timestamp	Expanded EHR
procvent	BreathType2	timestamp	Expanded EHR
procvent	Deltap	timestamp	Expanded EHR
procvent	elapsed_from_start	timestamp	Expanded EHR
procvent	FiO2	timestamp	Expanded EHR
procvent	FlowRate	timestamp	Expanded EHR
procvent	IERation	timestamp	Expanded EHR
procvent	InspPressure	timestamp	Expanded EHR
procvent	ITime	timestamp	Expanded EHR
procvent	MeanAirwayPressure	timestamp	Expanded EHR
procvent	MV	timestamp	Expanded EHR
procvent	nippv_id	timestamp	Expanded EHR
procvent	PEEP	timestamp	Expanded EHR
procvent	PIP	timestamp	Expanded EHR
procvent	PressureSupport	timestamp	Expanded EHR

procvent	Procedure	timestamp	Expanded EHR
procvent	ProcedureVentilatorPTA	timestamp	Expanded EHR
procvent	Rate	timestamp	Expanded EHR
procvent	RespRateM	timestamp	Expanded EHR
procvent	Sensitivity	timestamp	Expanded EHR
procvent	TV	timestamp	Expanded EHR
procvent	vent_before	timestamp	Expanded EHR
procvent	vent_id	timestamp	Expanded EHR
procvent	VentilatorAction	timestamp	Expanded EHR
procvent	VentilatorMode	timestamp	Expanded EHR
procvent	VentilatorMode1	timestamp	Expanded EHR
procvent	VentilatorMode2	timestamp	Expanded EHR
procvent	VentilatorVersion	timestamp	Expanded EHR
receiving	HospitalUnit	admission	Expanded EHR
receiving	ModetoRec	admission	Expanded EHR
receiving	ReceivingHospitalUnit	admission	Expanded EHR
receiving	ReceivingID	admission	Expanded EHR
receiving	ReceivingModeDescriptor	admission	Expanded EHR
receiving	ReceivingName	admission	Expanded EHR
receiving	ReceivingType	admission	Expanded EHR
receiving	RecHospDesignation	admission	Expanded EHR
receiving	ReferringModeDescriptor	admission	Expanded EHR
respiratory	BreathSoundsLeft	admission	Expanded EHR
respiratory	BreathSoundsRight	admission	Expanded EHR
respiratory	OxygenDeliveryMethod	admission	Expanded EHR
respiratory	OxygenFlow	admission	Expanded EHR
respiratory	RespEffort	admission	Expanded EHR
scene	AnatomicLocation	start	Expanded EHR
scene	AnatomicLocation1	start	Expanded EHR
scene	AnatomicLocation2	start	Expanded EHR
scene	AnatomicLocation3	start	Expanded EHR
scene	AnatomicLocation4	start	Expanded EHR
scene	AnatomicLocation5	start	Expanded EHR
scene	AnatomicLocation6	start	Expanded EHR
scene	AnatomicLocation7	start	Expanded EHR
scene	Belongings	start	Expanded EHR
scene	InitialPatientAcuity	start	Expanded EHR
scene	NumberPatientsAtScene	start	Expanded EHR
scene	SceneDescription	start	Expanded EHR

times	DateArrived_elapsed_from_start	timestamp	Expanded EHR
times	DateArriveRec_elapsed_from_start	timestamp	Expanded EHR
times	DateAtPt_elapsed_from_start	timestamp	Expanded EHR
times	DateAvailable_elapsed_from_start	timestamp	Expanded EHR
times	DateDispatched_elapsed_from_start	timestamp	Expanded EHR
times	DateEnroute_elapsed_from_start	timestamp	Expanded EHR
times	DateInQtrs_elapsed_from_start	timestamp	Expanded EHR
times	DateLeavePt_elapsed_from_start	timestamp	Expanded EHR
times	DateLeaveRef_elapsed_from_start	timestamp	Expanded EHR
times	DateNotified_elapsed_from_start	timestamp	Expanded EHR
times	DateofDescent_elapsed_from_start	timestamp	Expanded EHR
times	DateReceived_elapsed_from_start	timestamp	Expanded EHR
times	DateStandby_elapsed_from_start	timestamp	Expanded EHR
times	DateTxCare_elapsed_from_start	timestamp	Expanded EHR
ventilator	BreathType_vent	admission	Expanded EHR
ventilator	FIO2_vent	admission	Expanded EHR
ventilator	IERation_vent	admission	Expanded EHR
ventilator	Ino_vent	admission	Expanded EHR
ventilator	InspPressure_vent	admission	Expanded EHR
ventilator	ITime_vent	admission	Expanded EHR
ventilator	MeanAirwayPressure_vent	admission	Expanded EHR
ventilator	MV_vent	admission	Expanded EHR
ventilator	PEEP_vent	admission	Expanded EHR
ventilator	PIP_vent	admission	Expanded EHR
ventilator	PressureSupport_vent	admission	Expanded EHR
ventilator	PTA	admission	Expanded EHR
ventilator	Rate_vent	admission	Expanded EHR
ventilator	RespRateM_vent	admission	Expanded EHR
ventilator	TV_vent	admission	Expanded EHR
ventilator	VentMode	admission	Expanded EHR