Measures and Topics

Measures are entities used to quantify healthcare processes, outcomes, and patient survey results for hospitals, nursing homes, physicians, and clinical groups. Measures are mostly represented as rates in percentage and used in the reports for comparing various providers.

Organization

Similar measures are grouped into Topics, which are further grouped into Topic Categories. A measure may be assigned to multiple topics.

Measures, their Topics, and Topic Categories are directly related to the Targets that define them. They are imported into the MONAHRQ database along with a custom ***Error! Reference source not found.***'s other installation logic. They may be associated with one or more websites, and are consumed by ***Error! Reference source not found.****s*.

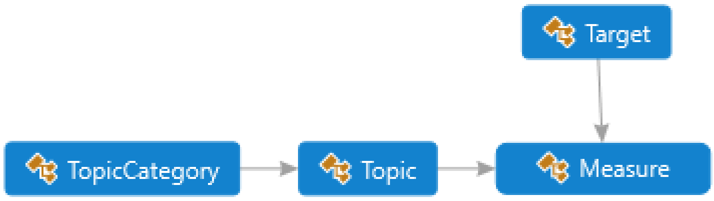

Figure 1 Relationship between Categories, Topics and Measures

Associations between measures and topics may also be changed by MONAHRQ Host Users from the "Measures" library.

Built-In Measures and Topics

There are more than 100 topics shipped with MONAHRQ. Most of these are maintained in a CSV file located in `Monahrq.Infrastructure\Resource\BaseData\MeasureTopics*.csv` and are loaded by the `TopicsStrategy`[1] class. See ***Error! Reference source not found.*** for more information on the import and update process for topics included in this CSV file.

Most built-in measures are installed as part of the installation process for built-in Wings, which is described in *Defining Custom Measures and Topics*

The specific steps for defining custom measures and topics varies depending on the method chosen to maintain those measures and topics. At a high level, the procedure is roughly the same:

1. Insert any missing TopicCategories
2. Insert any missing Topics
3. Insert or update Measures and associate them with your Topics and Target type

Installing into the MONAHRQ Database.

[1] See `Monahrq.Infrastructure.BaseDataLoader.Loaders.TopicsStrategy`, an implementation of `IBasedataImporter`

Defining Custom Measures and Topics

The specific steps for defining custom measures and topics varies depending on the method chosen to maintain those measures and topics. At a high level, the procedure is roughly the same:

4. Insert any missing TopicCategories
5. Insert any missing Topics
6. Insert or update Measures and associate them with your Topics and Target type

Installing into the MONAHRQ Database

Installation of Measures, their Topics and Topic Categories into the MONAHRQ database is handled by the Target's ***Error! Reference source not found.***. Although not required, this section assumes that your Module Definition inherits from `TargetedModuleWithMeasuresAndTopics<T>`.

The ImportMeasureTopics() Method

This method is intended to create or update database rows related to measure Topics and their Topic Categories. It is executed before `ImportMeasures()` as Measures must be associated with a Topic. Implementers of this method should query the database for required `TopicCategory` and `Topic` records, inserting any missing records or updating existing records as needed.

> `IMeasureService` provides simple methods for importing topics and measures from the filesystem. For information about the expected file format, refer to `Monahrq.Infrastructure.Entities.Domain.Measures.MeasureService`.

The ImportMeasures() Methods

This method is intended to create or update database rows related to individual Measures, if the required Topics have already been created. It is executed after `ImportMeasureTopics()`. Implementers of this method should query the database for required Measure records, inserting any missing records or updating existing records as needed.

```
var measure = session.Query<Measure>().FirstOrDefault(m => m.Name == "Sample Measure");
var target = session.Query<Target>().FirstOrDefault(
    t => t.Name == this.TargetAttribute.Name);
if (measure == null)
{
    measure = Measure.CreateMeasure(typeof(HospitalMeasure), target, "SMPL-1");
    measure.MeasureTitle.Plain = "My Sample Hospital Measure";
    measure.MeasureTitle.Clinical = "Sample Hospital Measure SMPL-1";
    measure.MeasureTitle.Policy = "Sample Hospital Measure SMPL-1";
    measure.AddTopic(topic);
    topic.Measures.Add(measure);
    session.Save(measure);
}
```

Figure 3 Sample Measure creation

The MeasureFilePath and MeasureTopicFilePath Properties

If you choose to define your measures and/or measure topics in the filesystem, these properties may be used to determine if measures and/or measure topics have changed[2] and need to be re-installed.

To use this functionality, return the path of your measure and/or measure topic files. To disable this functionality, return a null value from these properties.

Measure Types

Several different types of Measures are included with MONAHRQ

- CountyMeasure
- DynamicMeasure
- HospitalMeasure
- Measure
- NursingHomeMeasure
- PhysicianMeasure
- RegionMeasure

While different measure types receive slightly different treatment in the MONAHRQ Host User interface[3], there is no functional difference in the different types. For consistency, choose the most appropriate measure type for your purpose; if no type is directly applicable, use the Measure base class.

The chosen measure type implementation is passed as the first parameter to the `Measure.CreateMeasure(Type, Target, String)` method. The measure type is stored in the `Measures.ClassType` database column.

Measure Names / ID Codes

Each measure is given a unique, human readable, string identifier. In the example above, the ID is "SMPL-1"; this suggests that the measure belongs to the SMPL (or "Sample") module and is the first measure in that module. Other examples of measure IDs are "IQI 01" (inpatient quality indicators #1), "IP-10" (inpatient county median cost). There are no conventions or requirements for this code if it is supplied and unique.

The terms "measure name" and "measure ID" are sometimes used interchangeably.

[2] This determination is made by comparing the last write timestamp of the file to the `LastWingUpdate` column in the `Wing` table; for more information, refer to the `TargetedModuleWithMeasuresAndTopics<T>.RefreshDb()` source code

[3] Refer to `Monahrq.Measures.ViewModels.MeasureDetailsViewModel`, `Monahrq.Website.ViewModels.WebsiteEditMeasuresViewModel`, and `Monahrq.Website.ViewModels.WebsiteMeasuresViewModel`

Best Practices for Managing Measures and Topics

The recommended method for managing topics and measures is to store them in XML or CSV files, which are parsed during the installation process; these files may be stored in the filesystem or as embedded resources. Refer `Monahrq.Wing.Ahrq` for a sample using CSV files, and `Monahrq.Wing.HospitalSpendingSample` for a sample using XML files.

> ℹ The schema of these files is defined by the module(s) that read them. Although there is some consistency (i.e.: CSV and XML files from one module are very similar to CSV and XML files from another module), there is no standard.

User Interface

Measures and topics are exposed in the MONAHRQ Host User interface in the ***Error! Reference source not found.*** library, and in the “Modify Measures” tab of the ***Error! Reference source not found.*** library. In either location, measure names and descriptions