Splunk® Common Information Model Add-on Common Information Model Add-on Manual 4.11.0

How to use the CIM data model reference tables

Generated: 5/26/2018 4:18 am

How to use the CIM data model reference tables

Each topic in this section contains a use case for the data model, a breakdown of the required tags for the event datasets or search datasets in that model, and a listing of all extracted and calculated fields included in the model.

A **dataset** is a component of a data model. In versions of the Splunk platform prior to version 6.5.0, these were referred to as data model objects.

How to read the tags tables

The tags tables communicate which tags you must apply to your events in order to make them CIM-compliant. These tags act as **constraints** to identify your events as relevant to this data model, so that this data is included in Pivot reports, searches, and dashboards based on this model.

There might be additional constraints outside the scope of these tables. Refer to the data model itself using its editor view in Splunk Web for required fields, field=value combinations, or base searches that the model depends on.

Apply tags to your events to ensure your data is populated in the correct dashboards, searches, and Pivot reports.

1. Identify the CIM data model relevant to your events.
2. Identify the dataset within that model that is relevant to your events.
3. Observe which tags are required for that dataset.
4. Observe which tags are required for any parent datasets.
5. Observe any other constraints relevant to the dataset or its parents.
6. Apply those tags and other constraints to your events using event types.
7. Repeat for any additional relevant CIM datasets.

For a detailed walkthrough of these steps, see Use the CIM to normalize data at search time.

How to read the fields tables

The fields tables list the **extracted fields** and **calculated fields** for the event and search datasets in the model and provide descriptions and expected values (if relevant) for these fields.

How to find a field

The table presents the fields in alphabetical order, starting with the fields for the root datasets in the model, then proceeding to any unique fields for child datasets. The table does not repeat any fields that a child dataset inherits from a parent dataset, so refer to the parent dataset to see the description and expected values for that field.

Because the fields tables exclude inherited fields, many child datasets have no fields listed in the table at all. Those child datasets include only inherited fields from one or more of their parent datasets, so there are no unique extracted or calculated fields to display. All data models inherit the fields `_time`, `host`, `source`, and `sourcetype`, so those fields are always available to you for use in developing Pivot reports, searches, and dashboards.

How to interpret the expected values

For some fields, the tables include one or more expected values for that field. These expected values include:

- values that are used in knowledge objects in downstream applications such as Splunk Enterprise Security
- values that are used in the CIM model as constraints for a dataset

In some cases, the expected values also include additional values that Splunk suggests as the normalized standards for a field. The expected values are provided to help you make normalization decisions when developing add-ons. They are not exhaustive or exclusive.

Use the tables to apply the Common Information Model to your data

The tables in this section of documentation are intended to be supplemental reference for the data models themselves. Use the documentation and the data model editor in Splunk Web together. You can also access all of the information about a data model's dataset hierarchy, fields, field descriptions, and expected values in the JSON file of the model. You can browse the JSON in the `$SPLUNK_HOME/etc/apps/Splunk_SA_CIM/default/data/models` directory.

Prerequisite

You need Write access to a data model in order to browse it in its editor view. If you do not have this access, request it from your Splunk administrator.

Steps

1. In Splunk Web, go to **Settings > Data Models** to open the **Data Models** page.
2. Click a data model to view it in an editor view. There, you can see the full dataset hierarchy, a complete listing of constraints for each dataset, and full listing of all inherited, extracted, and calculated fields for each dataset.
3. Compare this information with the reference tables in the documentation for descriptions and expected values of the fields in each datasets.

	Information available in documentation	Information available in Data Model Editor in Splunk Web	Information available in JSON file of the model
Required tags	YES	YES	YES
Other constraints	NO	YES	YES
Full dataset hierarchy	NO	YES	YES
Inherited fields	NO	YES	YES
Extracted fields	YES	YES	YES
Calculated fields	YES	YES	YES
Data types	YES	YES	YES

Descriptions	YES	NO	YES
Expected values	YES	NO	YES
TA relevance	NO	NO	YES

How to access information directly from the JSON files

As shown in the table in the previous section, each data model's JSON file contains all the information about the model structure and its fields, so you can access this information programmatically. Several parameters formerly available only in the documentation are now available in the JSON's `comment` field. The format for this field is `{"description": "Description of the field.", "expected_values": ["val 1", "val 2"], "ta_relevant": true|false}`.

Parameter	Description
`description`	A description of the field.
`expected_values`	Optional. The values that applications such as Splunk Enterprise Security or Splunk App for PCI Compliance expect this field to contain. Use this for validation to ensure that your data populates correctly in the dashboards for these apps.
`ta_relevant`	Optional. A boolean indicator, signaling whether developers of add-ons need to populate this field. The default is true. A false value is given for fields that Splunk Enterprise Security or Splunk App for PCI Compliance automatically populate through the asset and identity correlation framework of those apps, or for other fields that are not intended to be populated by incoming data, such as the `tag` fields in each model.