Foglight® 8.1.0 **Dashboard Support Guide**

© 2025 Quest Software Inc.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept.
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. "Apache HTTP Server", Apache, "Apache Tomcat" and "Tomcat" are trademarks of the Apache Software Foundation. Google is a registered trademark of Google Inc. Android, Chrome, Google Play, and Nexus are trademarks of Google Inc. Red Hat, JBoss, the JBoss logo, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries. CentOS is a trademark of Red Hat, Inc. in the U.S. and other countries. Fedora and the Infinity design logo are trademarks of Red Hat, Inc. Microsoft, .NET, Active Directory, Internet Explorer, Hyper-V, Office 365, SharePoint, Silverlight, SQL Server, Visual Basic, Windows, Windows Vista and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. AIX, IBM, PowerPC, PowerVM, and WebSphere are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java, Oracle, Oracle Solaris, PeopleSoft, Siebel, Sun, WebLogic, and ZFS are trademarks or registered trademarks of Oracle and/or its affiliates in the United States and other countries. SPARC is a registered trademark of SPARC International, Inc. in the United States and other countries. Products bearing the SPARC trademarks are based on an architecture developed by Oracle Corporation. OpenLDAP is a registered trademark of the OpenLDAP Foundation. HP is a registered trademark that belongs to Hewlett-Packard Development Company, L.P. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries. Novell and eDirectory are registered trademarks of Novell, Inc., in the United States and other countries. VMware, ESX, ESXi, vSphere, vCenter, vMotion, and vCloud Director are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Sybase is a registered trademark of Sybase, Inc. The X Window System and UNIX are registered trademarks of The Open Group. Mozilla and Firefox are registered trademarks of the Mozilla Foundation. "Eclipse", "Eclipse Foundation Member", "EclipseCon", "Eclipse Summit", "Built on Eclipse", "Eclipse Ready" "Eclipse Incubation", and "Eclipse Proposals" are trademarks of Eclipse Foundation, Inc. IOS is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. Apple, iPad, iPhone, Mac OS, Safari, Swift, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Ubuntu is a registered trademark of Canonical Ltd. Symantec and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. OpenSUSE, SUSE, and YAST are registered trademarks of SUSE LCC in the United States and other countries. Citrix, AppFlow, NetScaler, XenApp, and XenDesktop are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. AlertSite and DéjàClick are either trademarks or registered trademarks of Boca Internet Technologies, Inc. Samsung, Galaxy S, and Galaxy Note are registered trademarks of Samsung Electronics America, Inc. and/or its related entities. MOTOROLA is a registered trademarks of Motorola Trademark Holdings, LLC. The Trademark BlackBerry Bold is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. Quest is not endorsed, sponsored, affiliated with or otherwise authorized by Research In Motion Limited. Ixia and the Ixia four-petal logo are registered trademarks or trademarks of Ixia. Opera, Opera Mini, and the O logo are trademarks of Opera Software ASA. Tevron, the Tevron logo, and CitraTest are registered trademarks of Tevron, LLC. PostgreSQL is a registered trademark of the PostgreSQL Global Development Group. MariaDB is a trademark or registered trademark of MariaDB Corporation Ab in the European Union and United States of America and/or other countries. Vormetric is a registered trademark of Vormetric, Inc. Intel, Itanium, Pentium, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Debian is a registered trademark of Software in the Public Interest, Inc. OpenStack is a trademark of the OpenStack Foundation. Amazon Web Services, the "Powered by Amazon Web Services" logo, and "Amazon RDS" are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries. Infobright, Infobright Community Edition and Infobright Enterprise Edition are trademarks of Infobright Inc. POLYCOM®, RealPresence® Collaboration Server, and RMX® are registered trademarks of Polycom, Inc. All other trademarks and registered trademarks are property of their respective owners.

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT NOTE, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

Foglight Dashboard Support Guide
Foglight Version - 8.1.0

Contents

1

About Dashboard Support Tools

Welcome to the Dashboard Support Guide. This guide describes the pages that you can use to evaluate your custom dashboards.

It describes the Tools and Dashboard Support page, explains how to view information about the data model, and shows you how to explore and publish the unit tests defined for module functions and queries. It contains information about exploring type and property definition inconsistencies, shows you how to view help identifiers per module, and explains where to find information on view usage in modules or sub-modules. Finally, it describes the process of viewing the amounts of data persisted to the database per dashboard, explains the concepts of data sources and their validation, and shows you how to review view error messages raised against one or more modules.

The Tools and Dashboard Support page is your starting point when you want to evaluate the dashboards you build for your monitored system. Use it to quickly assess the state of your dashboards and the underlying model, detect potential problems early on, and prevent unexpected behavior in the application.

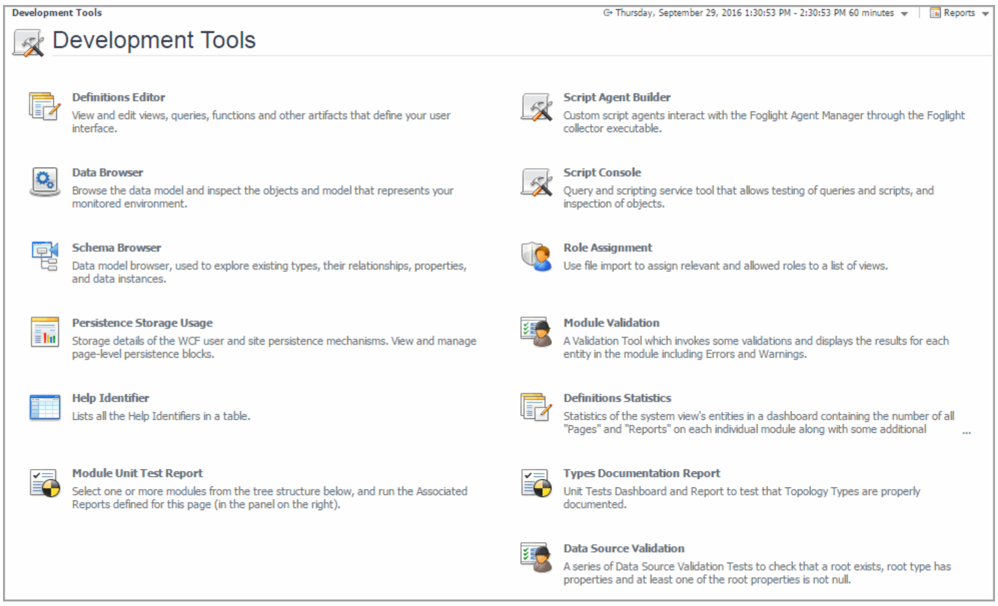
NOTE: The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

- Getting Started

Getting Started

To access this dashboard, on the navigation panel, click **Development Tools**.

Figure 1. Development Tools page



The Development Tools page contains links to dashboards that you can use to verify different aspects of your dashboards during the development cycle. For complete information, see the following chapters:

- Definitions Editor: For more information, see the “Definitions Pane” section in the Foglight Web Component Guide.
- Data Browser: For more information, see the “Data and Data Sources Pages” section in the Foglight Web Component Guide.
- Schema Browser: For more information, see Exploring the Data Model on page 7.
- Persistence Storage Usage: For more information, see Viewing and Managing Persistence Storage Usage on page 26.
- Help Identifier: For more information, see Finding Help Identifiers on page 19.
- Module Unit Test Report: For more information, see Reporting on Module Unit Tests on page 12.
- Script Agent Builder: For more information, see the “Building Script Agents” section in the Foglight Administration and Configuration Guide.
- Script Console: For more information, see the “Retrieving Data with Scripts and Queries” section in the Foglight Administration and Configuration Guide.
- Role Assignment: For more information, see the “Explore the Roles tab” section in the Foglight Administration and Configuration Guide.
- Module Validation: For more information, see Validating Modules on page 28.
- Definitions Statistics: For more information, see Exploring Module Definitions on page 21.
- Types Documentation Report: For more information, see Exploring Type and Property Definition Inconsistencies on page 17.
- Data Source Validation: For more information, see Validating Data Sources on page 31.

2

Exploring the Data Model

Foglight® collects data from the monitored system and uses that data to dynamically build topology models at run-time. A topology model organizes the data in a way that represents the logical and physical relationship between items in your monitored environment and provides the context for the collected metrics.

Topology models consist of nodes, where each node is an instance of a topology object. The nature of the monitored environment dictates the structure and complexity of each topology model and the collection of available topology types. A basic server installation includes a set of core topology types, and each installed cartridge adds to that collection.

Use the Schema Browser to view information about the available data types, their relationships in the data model, properties, and object instances. This dashboard can help you better understand the data model structure.

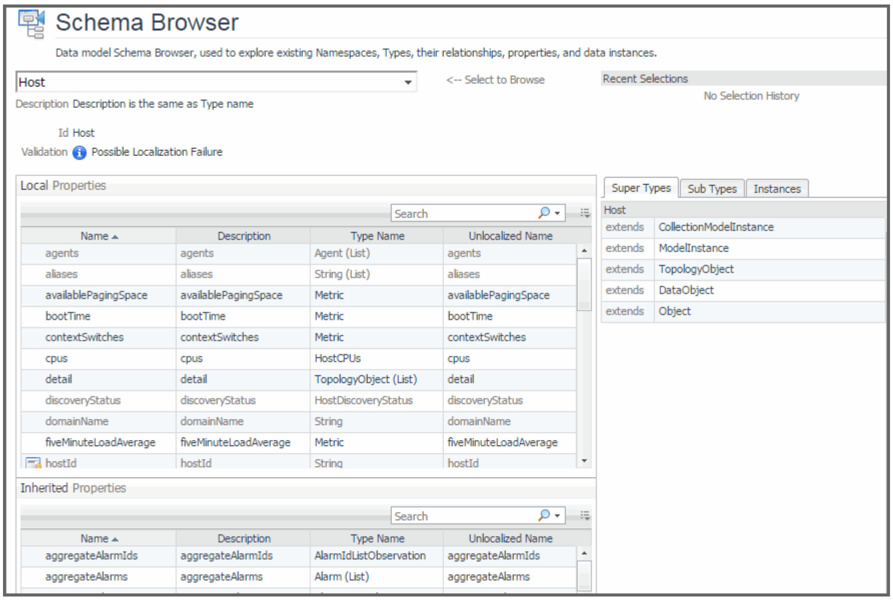
> **NOTE:** The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

- Getting Started
- Drilling Down on Types

Getting Started

To access this dashboard, on the Development Tools page, click **Schema Browser**.

Figure 1. Schema Browser



Drilling Down on Types

The Schema Browser allows you better understand the database schema. Start by selecting a namespace. A namespace is a collection of types that share the same concept. For example, the `Administration/Setup & Support/Blackouts` namespace contains a group of types that are used to implement blackouts in Foglight. While type names must be unique within a namespace, it is possible to have same type names across multiple namespaces.

Review the list of types associated with the namespace and drill down on a specific type to review its properties, instances, and relationships with other types in the model.

To view a list of types associated with a namespace:

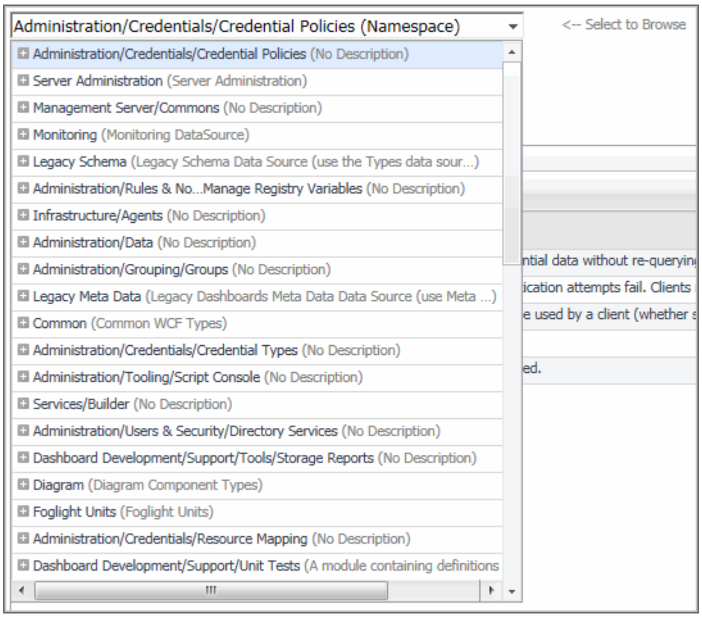
1 On the Development Tools page, click **Schema Browser**.

The Schema Browser dashboard appears in the display area.

2 Select a namespace.

In the Schema Browser dashboard, click the **Select to Browse** box.

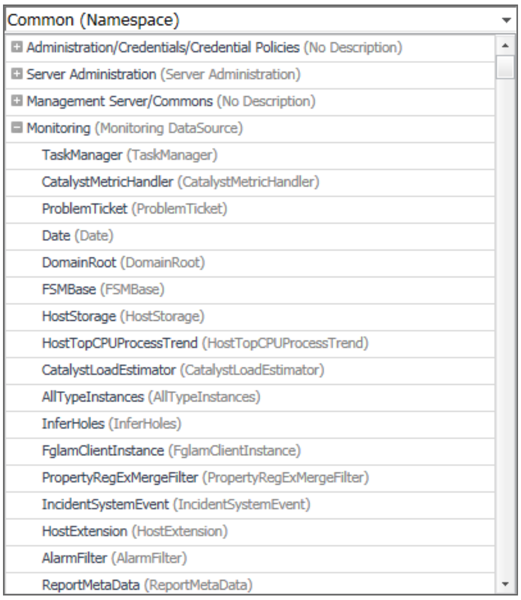
A list of nodes appears in the tree view, with each node representing a namespace.



IMPORTANT: The type and collection of available namespaces depends on the range of installed cartridges.

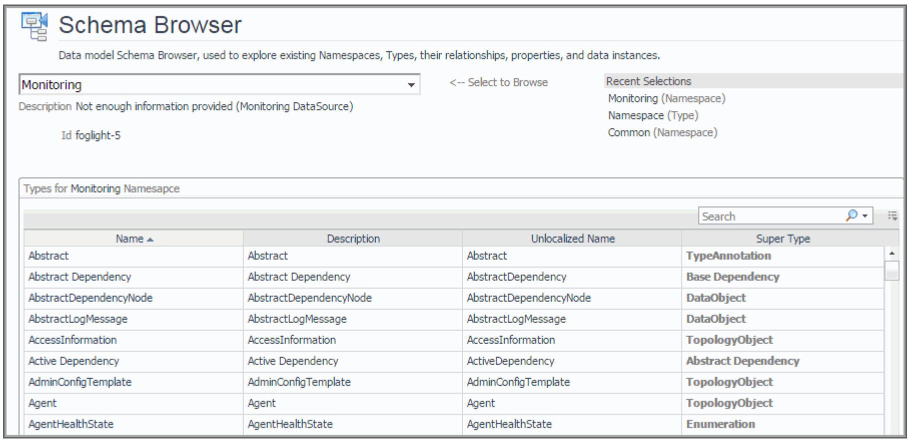
3 Expand a node in the list.

A list appears, showing the topology types associated with the selected namespace.



4 Click a namespace node in the list.

The Schema Browser dashboard refreshes, showing a table containing the types associated with the selected namespace. For each type, it shows its name, description, non-localized name, and the parent type.

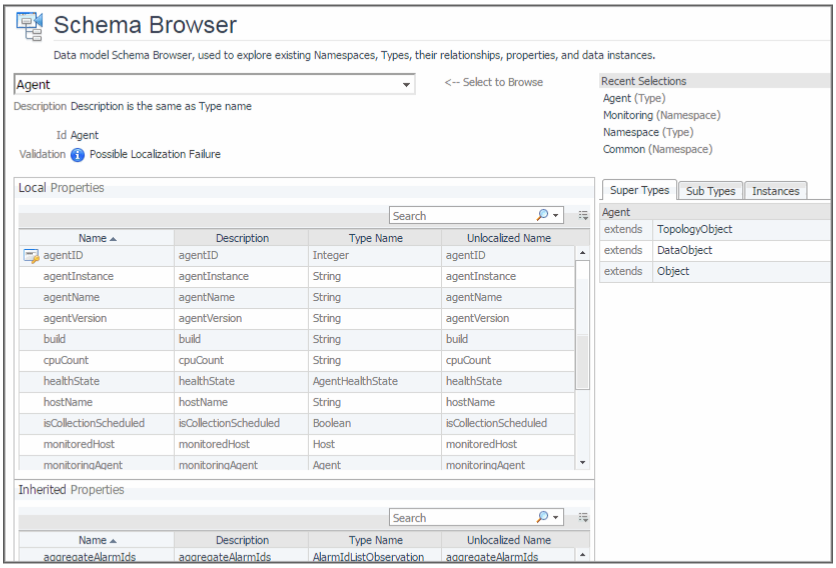


5 In the Schema Browser dashboard, drill down on a type by clicking its entry in the table.

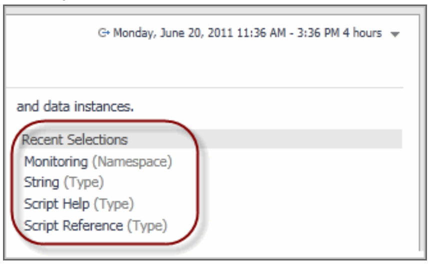
TIP: Another way to drill down on a type is to click a type entry in the **Select to Browse** list.

IMPORTANT: Selecting a generic type, for example, String, does not show any information about that type in the Schema Browser. The Schema Browser only displays information about Foglight® types.

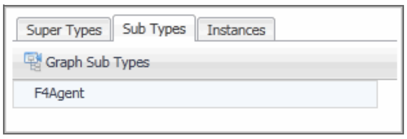
The Schema Browser dashboard refreshes, showing additional information about the selected type.



TIP: Drilling down on a type causes the Schema Browser to refresh and reflect information pertinent to the type. To quickly return to the previous view, use the links in the **Recent Selections** area.



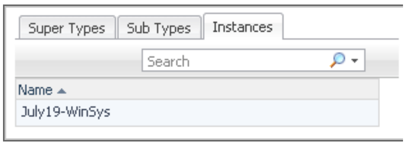
- **Local Properties** and **Inherited Properties**: Each table shows a list of properties that are local to the selected type (**Local Properties**) or inherited from a parent type (**Inherited Properties**). For each property, it indicates if the property is an identity property, and shows its name, description, type name, and non-localized name. Identity properties are indicated with the icon appearing on the left of the property name.
- **Super Types**: This tab shows a list of this type's parent (ancestor) types.
- **Sub Types**: This tab shows a list of this type's child (descendant) types.



Clicking **Graph Sub Types** shows a graphical representation of the child type's relationship with this type.

TIP: Use the breadcrumb trail to return to the Schema Browser.

- **Instances**: This tab shows a list of this type's instances.



6 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

Reporting on Module Unit Tests

The Module Unit Test Report lists the modules that exist in your environment. Each Foglight® module contains a collection of operational elements such as views, dashboards, sub-modules, and other entities. For each module, this dashboard shows the number of functions, queries, along with the numbers and percentages of unit tests defined for the module's functions and queries.

A unit test is a block of code that can be used to test the logic of a function or a query, to ensure it suits its purpose. Unit tests can detect potential code problems early in the development cycle. The higher percentage of unit tests in a module can prevent unexpected behavior in the application and, as such, improve its reliability.

Drill down on the unit test details to retrieve additional information about a module's unit test. To publish information about unit tests per module, run a report against one or selected modules using the reporting option from the action panel.

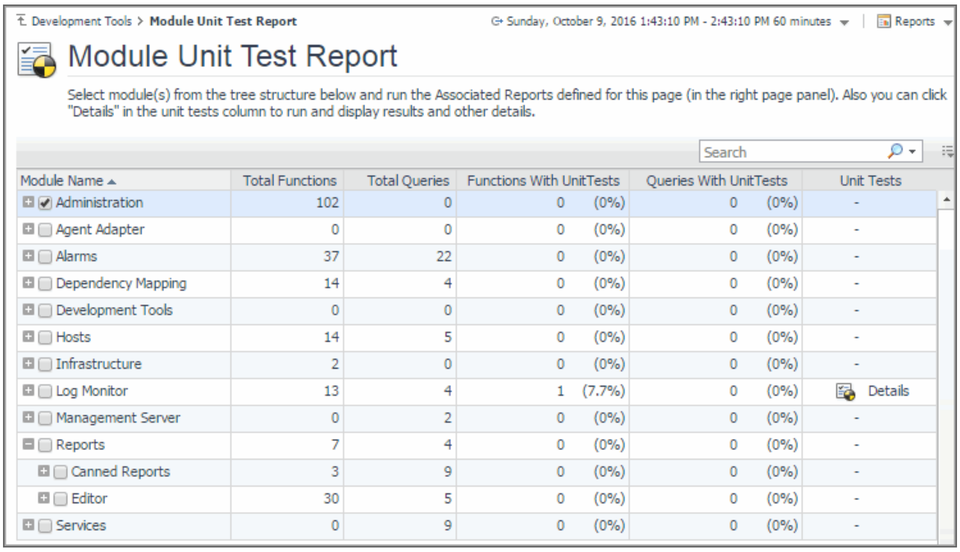
NOTE: The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

- Getting Started
- Exploring Module Unit Tests Result
- Publishing Module Unit Tests Result to a Report

Getting Started

To access this dashboard, on the Development Tools page, click **Module Unit Test Report**.

Figure 1. Module Unit Test Report



Module Name	Total Functions	Total Queries	Functions With UnitTests	Queries With UnitTests	Unit Tests
Administration	102	0	0 (0%)	0 (0%)	-
Agent Adapter	0	0	0 (0%)	0 (0%)	-
Alarms	37	22	0 (0%)	0 (0%)	-
Dependency Mapping	14	4	0 (0%)	0 (0%)	-
Development Tools	0	0	0 (0%)	0 (0%)	-
Hosts	14	5	0 (0%)	0 (0%)	-
Infrastructure	2	0	0 (0%)	0 (0%)	-
Log Monitor	13	4	1 (7.7%)	0 (0%)	Details
Management Server	0	2	0 (0%)	0 (0%)	-
Reports	7	4	0 (0%)	0 (0%)	-
Canned Reports	3	9	0 (0%)	0 (0%)	-
Editor	30	5	0 (0%)	0 (0%)	-
Services	0	9	0 (0%)	0 (0%)	-

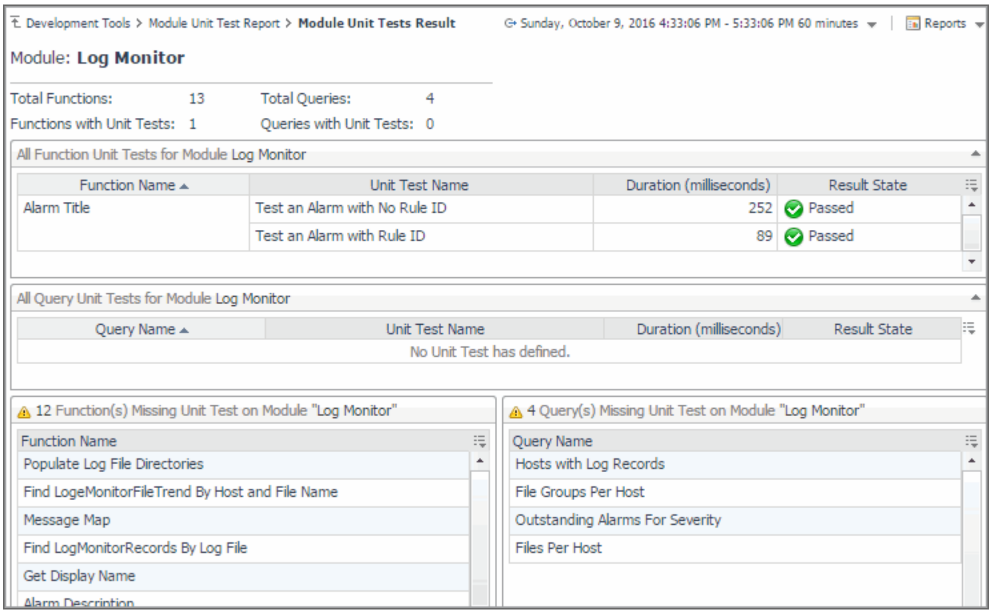
Exploring Module Unit Tests Result

If a module includes function of query unit tests, you can drill down on the unit test details to see the unit test details, whether they passed with success, or failed with errors. This information appears in the Module Unit Tests Result view. Display this view by drilling down on **Details** in the **Unit Tests** column on the Module Unit Test Report dashboard.

The Module Unit Tests Result view shows how many functions and queries exist in a module, the number of function and query unit tests for that module, the unit test results, and whether unit tests are defined for all functions and queries.

To return to the Module Unit Test Report, use the breadcrumb trail.

Figure 2. Returning to the Module Unit Test Report

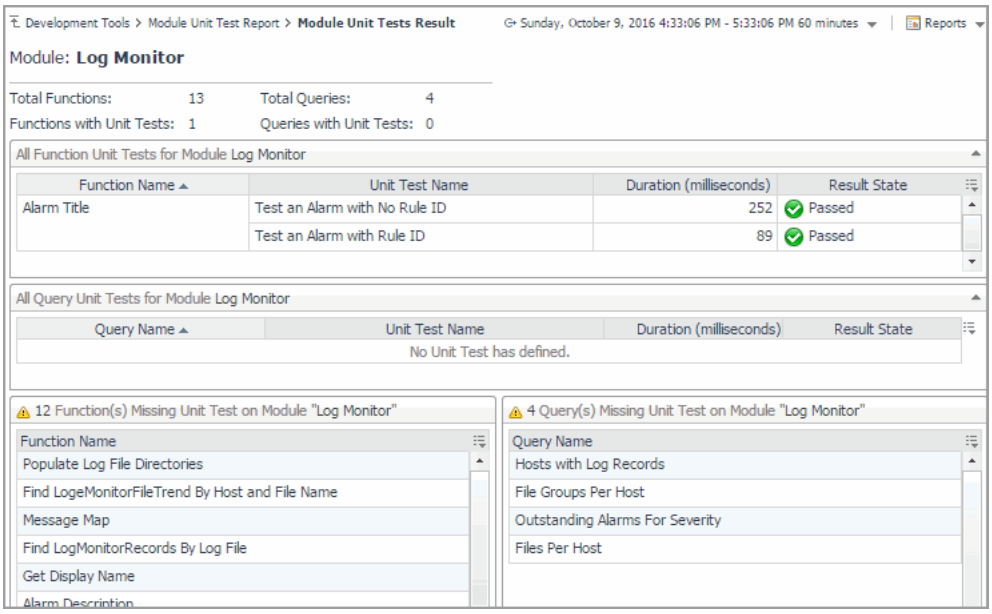


If any unit tests result in errors, this is indicated in the **Result State** column. Clicking this column shows additional information about the error.

To display module unit test result:

1. On the Development Tools page, click **Module Unit Test Report**.

   The Module Unit Test Report dashboard appears in the display area.
2. Observe the tree view of modules and the available unit tests per module.
3. Locate a node in the navigation tree by expanding its nodes to navigate to the module that includes unit tests.
4. In the **Unit Tests** column, click **Details**.
5. The display area refreshes, showing the Module Unit Tests Result view.



6 Observe the Module Unit Tests Result view.

The view shows how many functions and queries exist in a module, the number of function and query unit tests for that module, the unit test duration, results, and whether unit tests are defined for all functions and queries.

7 If any unit tests result in errors, click the **Result State** column to inspect the error message.

The **Error Details** message box appears, showing the error details.

8 To return to the Module Unit Test Report dashboard, use the breadcrumb trail in the display area.

9 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

Publishing Module Unit Tests Result to a Report

Foglight® comes with report templates that you can use to publish module unit tests result to a report. Depending on the level of information that you want to see in the report, you can report on all modules, or selected modules only.

Reporting on all modules results in a flat list of all modules and sub-modules showing the same information that appears on the Module Unit Test Report. For each module and sub-module, it shows the number of functions, queries, along with the numbers and percentages of function and query unit tests.

Reporting on selected module shows more details about unit tests. For example, you can see the numbers of all functions and queries compared to the numbers of the module's functions and queries that include unit tests. For each unit test, you can see its name, duration, and result, and a list of functions or queries that are missing unit tests. Use this information as a quick reference to investigate unit test results and to identify which components are missing unit tests.

To publish module unit tests to a report:

1 On the Development Tools page, click **Module Unit Test Report**.

The Module Unit Test Report dashboard appears in the display area.

2 Observe the tree view of modules and the available unit tests per module.

3 To drill down on a module's sub-modules, expand the module node.

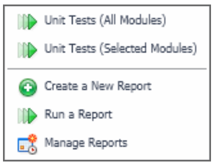
For each module that contains unit tests, you can see the numbers of functions, queries, and the associated unit tests.

4 Publish unit test results to a report.

a To create a report that displays unit test results for specific modules only, select those modules in the list.

b In the top-right corner of the display area, click **Reports**.

A dwell appears.



TIP: Another way to access this dwell is through the action panel. On the General tab, under Other Actions, click Create report.

c To report on all modules, in the dwell, click **Unit Tests (All Modules)**.

d To report on the modules selected in Step a, click **Unit Tests (Selected Modules)**.

TIP: Reporting on module unit tests takes the list of selected modules as input. When reporting on selected modules, it is possible to edit that list to add or remove modules, as required.

The **Schedule Report** dialog box appears.

e Complete the flow to schedule the report.

The newly generated reports can be accessed from the Reports dashboard. For more information about reports, see the *Foglight User Help* or click the **Help** button in the dialog box.

5 To return to the Development Tools page, click **Development Tools** in the breadcrumb trail.

4

Exploring Type and Property Definition Inconsistencies

The Types Documentation Report lists all namespaces that exist in your environment. A namespace is a collection of types that share the same concept. For example, each Foglight® module has a collection of types associated with it. While type names must be unique within a namespace, it is possible to have same type names across multiple namespaces. For each namespace, the list shows the types it contains and their properties, and whether their names follow the standard naming convention, and indicates if they include descriptions and enumeration descriptions (for `Enum` types) that are identical to their names.

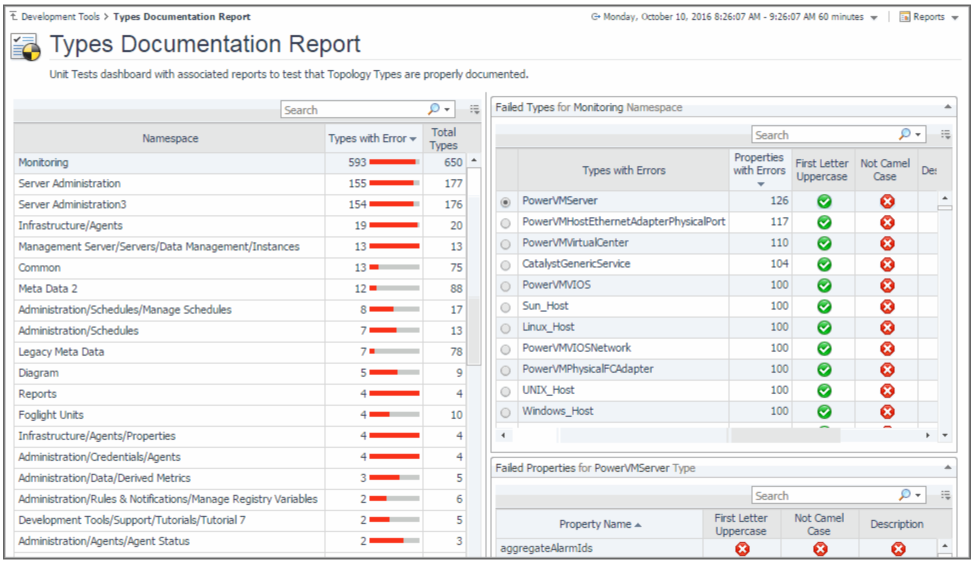
Use this dashboard to quickly evaluate if your model includes any types or properties with inconsistences in their name and description and to list them. Start by selecting a namespace that includes errors. Then, review the types that contain name or definition errors. For each type or property, the table shows its name, the number of properties with errors (types only), whether the first letter is an uppercase letter, the type name is a compound word with each element's initial letter capitalized, and if the description (and enumeration description for `Enum` types) exists or is the same as the type name.

> **NOTE:** The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

Getting Started

To access this dashboard, on the Development Tools page, click **Types Documentation Report**.

Figure 1. Types Documentation Report



To explore type and property definition inconsistencies:

1 On the Development Tools page, click **Types Documentation Report**.

The Types Documentation Report dashboard appears in the display area.

2 Observe the list of namespaces on the left.

For each namespace, the table shows the number of types with errors, and the total number of types.

3 Click a namespace in the table.

The **Failed Types** and **Failed Properties** views on the right refresh, showing the selected namespace types and properties with errors.

4 In the **Failed Types** view, observe the type errors.

For each type, the table shows its name, the number of properties with errors, whether the first letter is an uppercase letter, the type name is a compound word with each element's initial letter capitalized, and if the description (or enumeration description for `Enum` types) exists or is the same as the type name.

5 In the **Failed Types** view, click a type.

The **Failed Properties** view refreshes.

6 In the **Failed Properties** view, observe the property errors.

For each property, the table shows its name, whether the first letter is an uppercase letter, the type name is a compound word with each element's initial letter capitalized, and if the description exists or is the same as the type name.

7 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

5

Finding Help Identifiers

The Help Identifiers dashboard allows you to list help identifiers for a selected module or sub-module. Help identifiers can be embedded in help files to enable context-sensitive help for a specific component in the browser interface.

Start by selecting a module. For each help identifier, the list shows the related component name, its help ID, and the component purpose.

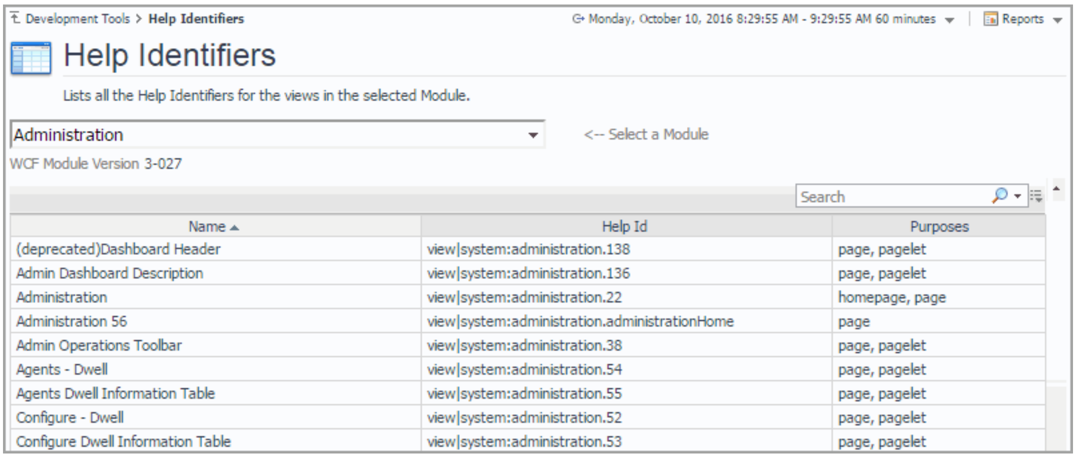
NOTE:

NOTE: The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

Getting Started

To access this dashboard, on the Development Tools page, click **Help Identifiers**.

Figure 1. Help Identifiers



Name	Help Id	Purposes
(deprecated)Dashboard Header	view|system:administration.138	page, pagelet
Admin Dashboard Description	view|system:administration.136	page, pagelet
Administration	view|system:administration.22	homepage, page
Administration 56	view|system:administration.administrationHome	page
Admin Operations Toolbar	view|system:administration.38	page, pagelet
Agents - Dwell	view|system:administration.54	page, pagelet
Agents Dwell Information Table	view|system:administration.55	page, pagelet
Configure - Dwell	view|system:administration.52	page, pagelet
Configure Dwell Information Table	view|system:administration.53	page, pagelet

To list help identifiers associated with a module:

1 On the Development Tools page, click **Help Identifiers**.

The Help Identifiers dashboard appears in the display area.

2 On the Help Identifiers dashboard, select a module.

The list of help identifiers refreshes, showing the help identifiers that are associated with the selected module.

3 Observe the list of help identifiers.

For each help identifier, the list shows the related component name, its help ID, and the component purpose.

4 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

Exploring Module Definitions

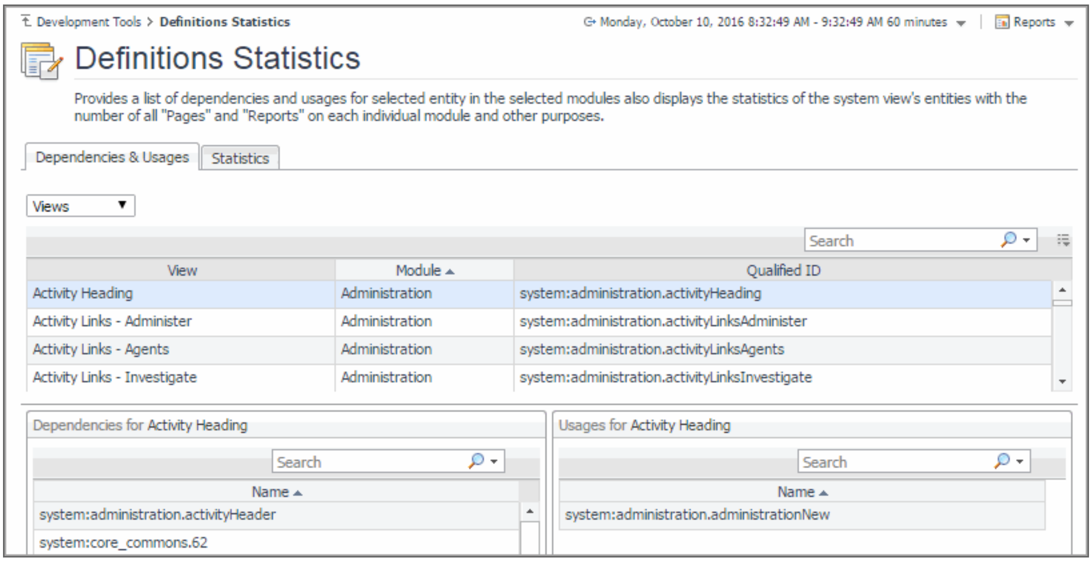
Web Component Framework (WCF) definitions such as views and queries in Foglight® are organized into modules and sub-modules. A module is a collection of related WCF definitions. Each module in Foglight contains a collection of operational elements such as views, dashboards, sub-modules, and other entities. The Definitions Statistics dashboard gives a quick overview of how views are used in individual modules or sub-modules. It also shows the relationship of components such as views or associations with other components.

NOTE: The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

Getting Started

To access this dashboard, on the Development Tools page, click **Definitions Statistics**. From there, start by selecting one or more modules in the **Module Selector** on the navigation panel.

Figure 1. Definitions Statistics

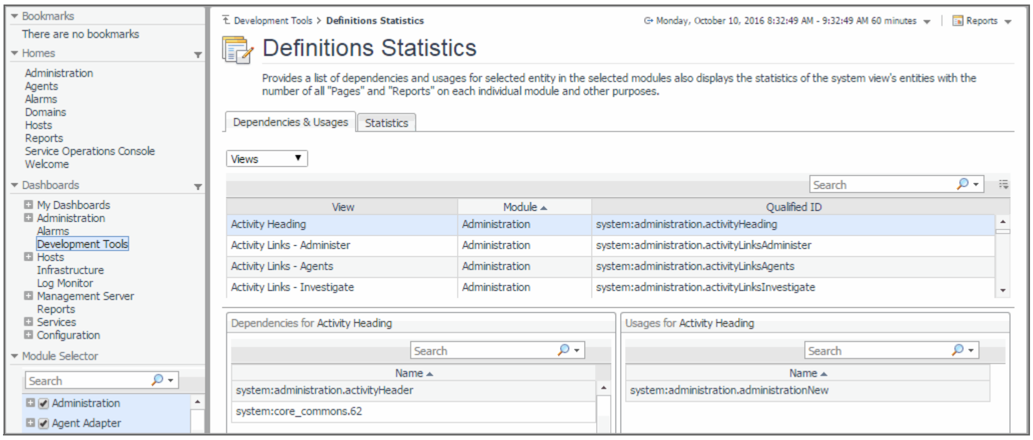


Exploring Component Dependencies and Usage

WCF definitions can contain nested WCF definitions. For example, a dashboard can contain a several layers of nested views. Container views *depend* on nested views to retrieve or display information provided by the nested views. In that sense, container views *use* nested components. Exploring these relationships can help you better understand the overall design of the browser interface, improve the efficiency of your development cycles by creating reusable building blocks, and simplify the collection of dashboard artifacts by removing obsolete components that are no longer needed or referenced.

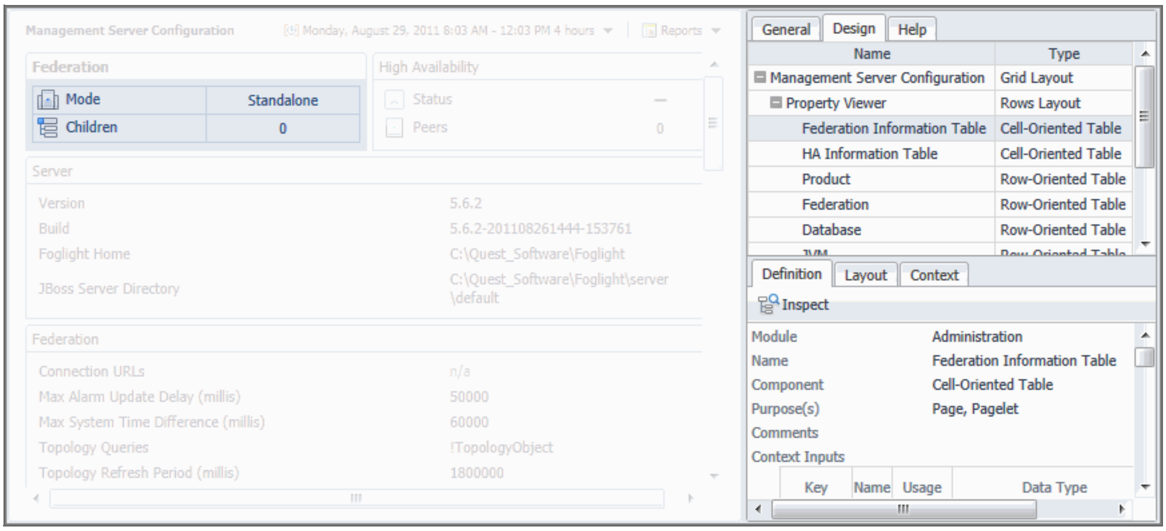
The **Dependencies & Usages** tab on the Definition Statistics dashboard helps you explore the relationships between the components in the selected module.

Figure 2. Dependencies & Usages tab on the Definition Statistics dashboard



Another way to view these relationships is by using the **Design** tab of the action panel. However, while the Definition Statistics dashboard shows only the components that are related to the selected component and defined in a selected module, the **Design** tab shows the related components regardless of the module in which they are defined.

Figure 3. Design tab on the action panel



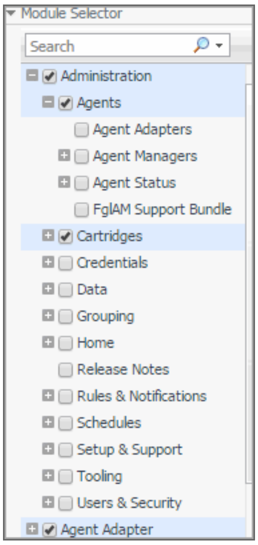
To explore component dependencies and usage:

1 On the Development Toolspage, click **Definitions Statistics**.

The Definitions Statistics dashboard appears in the display area and the **Module Selector** appears on the navigation panel.

2 On the navigation panel, expand a node in the **Module Selector**.

A list of sub-modules appears.

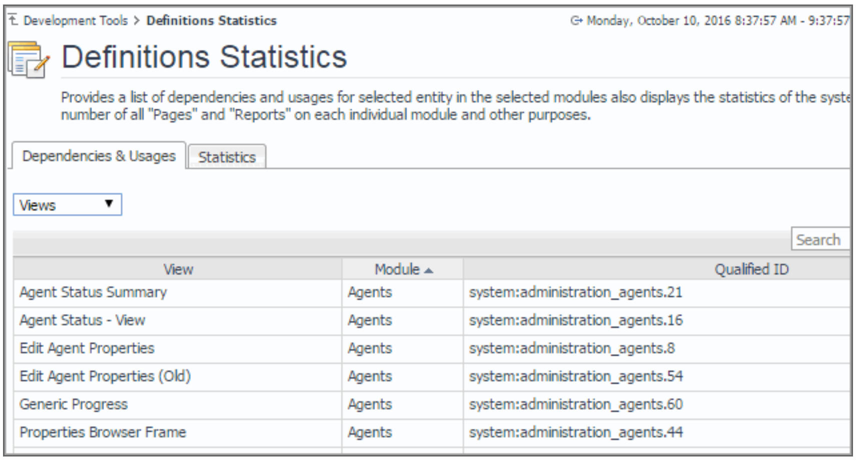


IMPORTANT: The type and collection of available modules depend on the cartridges installed.

TIP: Selecting a sub-module in the tree allows you to narrow down the statistics for a specific set of views in a module.

3 In the **Module Selector**, select one or more modules or sub-modules containing the components whose dependencies and usages you want to explore. For example, to explore the views included in the Setup & Support module, in the **Module Selector**, under Administration, select **Setup & Support**.

The **Dependencies & Usages** tab refreshes, showing the views that are defined in the selected module.



4 Drill down on a view to see its dependencies and usages. In the list of views, select a view.

For example, in the **View** column, scroll down and click **Property Viewer**.

The **Dependencies & Usages** tab refreshes.

The **Dependencies** view shows one nested view. The **Usages** view indicates that the selected view is part of other container views.

> **IMPORTANT:** A selected view can include additional nested views and be part of other container views that do are not listed here. That is because these views only list the components that are both related to the selected component and defined in the selected module.

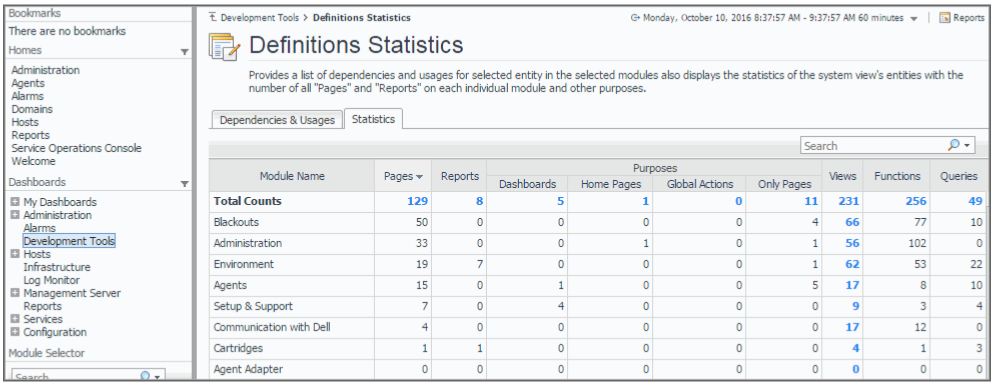
5 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

Viewing Module Definition Statistics

A view can be a page, dashboard, home page, or have a combination of these purposes. For each module, the list shows the numbers of views that are defined as pages, reports, dashboards, home pages, global actions, pages (only), and the total number of views. The ratio of different view purposes in a module can reflect its complexity given its number of home pages, dashboards, and reports. It can also give you an idea on the work required to develop and maintain a module.

The **Statistics** tab on the Definition Statistics dashboard displays a summary of how views are used in individual modules or sub-modules.

Figure 4. Statistics tab



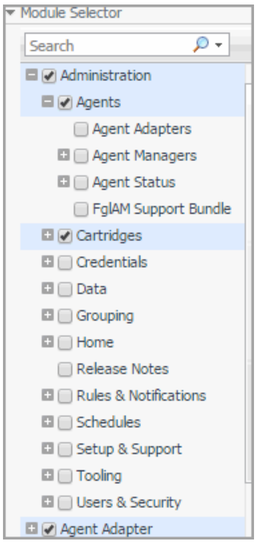
To see module definition statistics:

1 On the Development Tools page, click **Definitions Statistics**.

The Definitions Statistics dashboard appears in the display area and the **Module Selector** appears on the navigation panel.

2 On the navigation panel, expand a node in the **Module Selector**.

A list of sub-modules appears.

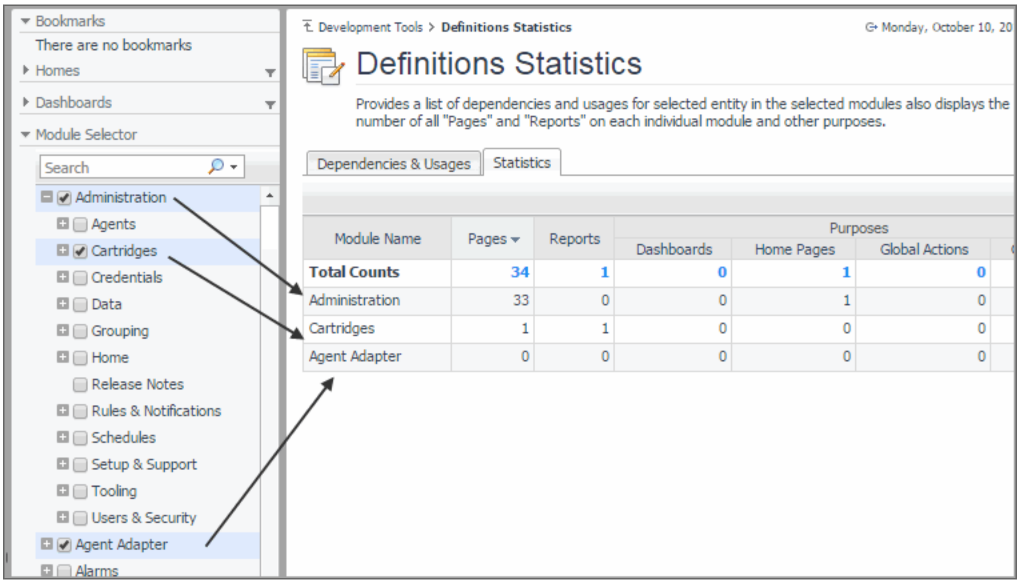


IMPORTANT: The type and collection of available modules depend on the range of installed cartridges.

TIP: Selecting a sub-module in the tree allows you to narrow down the statistics for a specific set of views in a module.

3 In the **Module Selector**, select all modules and sub-modules whose view definition statistics you want to see.

The Definitions Statistics dashboard refreshes, showing the views defined in the selected modules.



4 Observe the view definition statistics.

For each module, the Definitions Statistics dashboard shows the numbers of views that are defined as pages, reports, dashboards, home pages, global actions, pages (only), and the total number of views.

5 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

7

Viewing and Managing Persistence Storage Usage

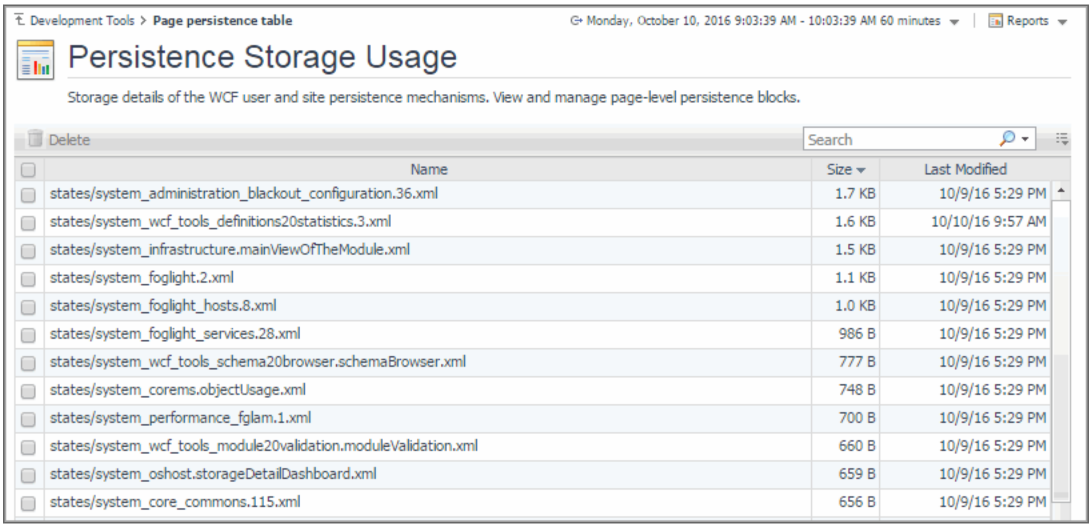
The Persistence Storage Usage dashboard allows you to view the amount of data persisted to the database for each individual dashboard, and the size of these data blocks. For each data block, it shows its size and the date and time it was last modified. You can delete these data blocks to restore the dashboards to their original state, as required. Data blocks are stored in the database. Sorting them by size tells you which data blocks are utilizing highest amounts of database resources which may affect the overall database performance, and delete them, as required, to improve the database performance.

NOTE: The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

Getting Started

To access this dashboard, on the Development Tools page, click **Persistence Storage Usage**.

Figure 1. Persistence Storage Usage



NOTE: The type and collection of data blocks depends on the range of installed cartridges.

To view and manage persistence storage usage:

1 On the Development Tools page, click **Persistence Storage Usage**.

The Persistence Storage Usage dashboard appears in the display area.

2 Observe the view definition statistics.

For each data file, the Persistence Storage Usage dashboard shows its size and the date and time it was last modified.

3 To delete one or more data files, select them in the table and click **Delete**.

The **Confirm File(s) Delete** dialog box appears.

4 To proceed, click **Delete**.

5 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

Validating Modules

Each Foglight® module contains a collection of operational elements such as views, dashboards, sub-modules, and other entities. Use the Module Validation dashboard to validate a module or sub-module and review the related messages. For each selected module or sub-module, the list shows the total number of error, warning, and information messages, and then the number of error, warning, and information messages for each entity in the module. Drilling down on a message number shows the validation message in a separate dialog box. Use this information to improve module definitions as you develop.

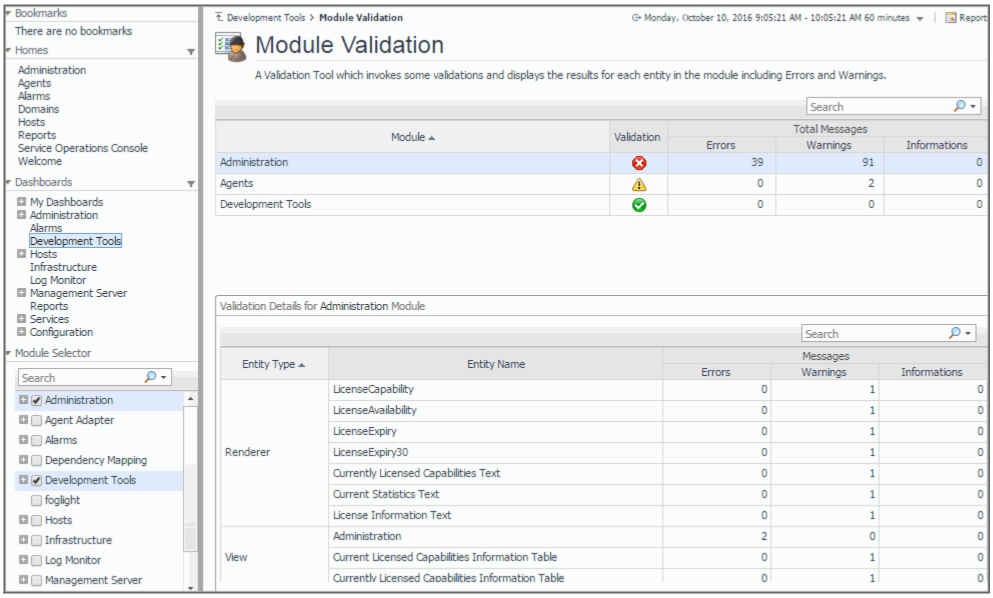
NOTE: If the Show deprecated definitions user preference setting is enabled, the numbers of messages related to deprecated components appear as a separate number in the list, and are marked Deprecated. To access the User Preferences, on the navigation panel, under Dashboards, select Configuration > User Preferences. For complete information about configuring user preferences, see the Foglight User Guide.

NOTE: The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

Getting Started

To access this dashboard, on the Development Tools page, click **Module Validation**. From there, start by selecting one or more modules or sub-modules in the **Module Selector** on the navigation panel.

Figure 1. Module Validation



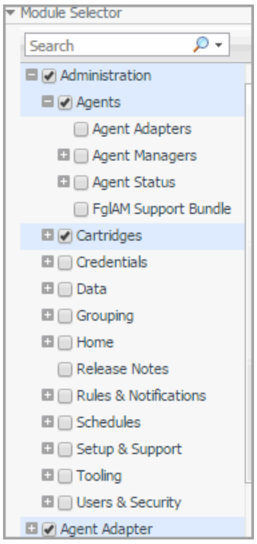
To validate a module:

1 On the Development Tools page, click **Module Validation**.

The Module Validation dashboard appears in the display area.

2 On the navigation panel, expand a node in the **Module Selector**.

A list of sub-modules appears.

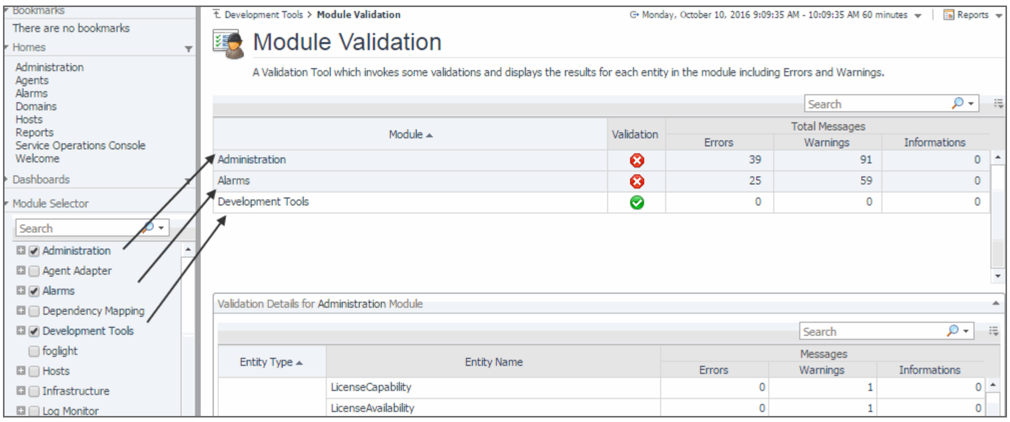


IMPORTANT: The type and collection of available modules depend on the range of installed cartridges.

TIP: Selecting a sub-module in the tree allows you to narrow down the statistics for a specific set of views in a module.

3 In the **Module Selector**, select all modules and sub-modules that you want to validate.

The Module Validation dashboard refreshes, showing the views defined in the selected modules.



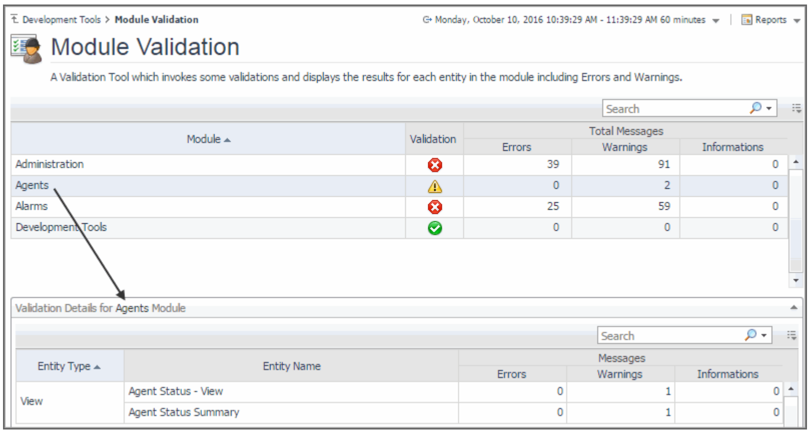
4 Observe the module validation statistics.

For each selected module or sub-module, the top table shows the validation state (Normal, Warning, or Fatal), and the number of error, warning, or informational messages associated with that component.

5 View detailed validation results for a specific module or sub-module.

a In the top table, click a row containing a module or sub-module.

On the Module Validation dashboard, the **Validation Details** table refreshes, listing the individual module entities against which one or more validation messages are generated.

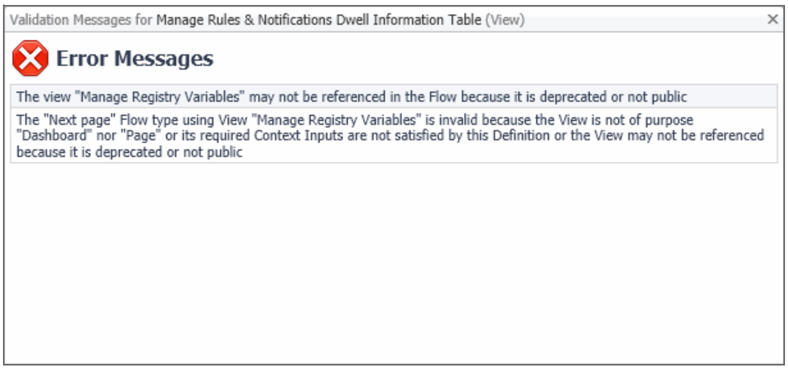


b Observe the **Validation Details** table.

For each entity, the table shows its type, name, and the numbers of error, warning, and informational messages.

c Review an entity's error, warning, or informational messages by clicking the related columns.

A dialog box appears, listing the messages of the selected type that are associated with the entity.



d Use this information to improve module definitions as you develop.

6 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

Validating Data Sources

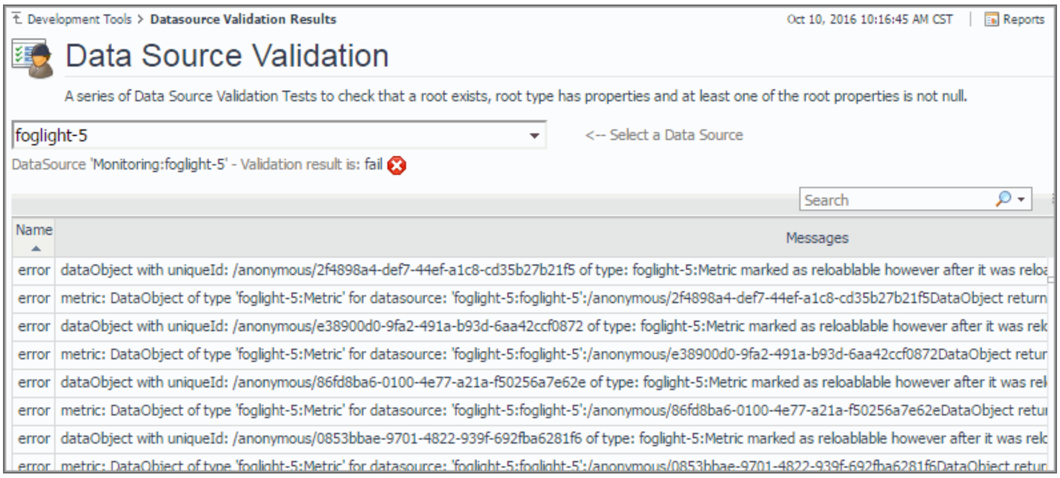
A data source contains information that is displayed in Foglight®. This information is stored in data objects as object properties. A data source encapsulates all the system knows about the collected data. It is organized as a dynamic graph of object types, starting from a root that represents the entire data model. The Data Source Validation dashboard allows you to select a data source and check if that data source has a root, and that the root type has properties defined. Detecting and solving data source problems early on can prevent problems in the browser interface. If any errors are found on the data source, they appear in the list. For each error, the list shows its type and the error description. The overall data source validation result also appears on the dashboard. Use this information to improve data source definitions as you develop. Warning messages usually appear for information purposes only, to indicate a non-fatal conditions. Unlike fatal messages, they do not prevent the data source from being validated.

NOTE: The Cartridge Developer role grants access to this dashboard. Your Foglight account must have this role to access this dashboard.

Getting Started

To access this dashboard, on the Development Tools page, click **Data Source Validation**.

Figure 1. Data Source Validation



To validate a data source:

1. On the Development Tools page, click **Data Source Validation**.

   The Data Source Validation dashboard appears in the display area.

2. On the Data Source Validation dashboard, select a data source.

   If any validation errors exist for the selected data source, they appear in the list.

3 Observe the list of data source validation errors.

   For each help identifier, the table shows its type and description. The overall validation result also appears just above the table. You can use this information to improve data source definitions, as required.

4 To return to the Development Tools page, click **Development** Tools in the breadcrumb trail.

About Us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit https://www.quest.com/company/contact-us.aspx or call +1-949-754-8000.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.