

© 2023 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 Where next meets now 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.

Contents

| | |
|---|-----------|
| The Foglight data model | 8 |
| What are models? | 8 |
| Who needs to know about models? | 9 |
| Modeling process overview | 9 |
| Where can I see the models? | 10 |
| Topology type definitions in XML | 11 |
| Schema browser | 11 |
| Data dashboard | 11 |
| How are models organized? | 12 |
| Where can I see the data model hierarchy? | 12 |
| What internal models are created? | 13 |
| What other models are created? | 13 |
| What is in “Model Roots”? | 13 |
| How are Foglight 4 agents represented? | 13 |
| Why might I need to navigate models? | 14 |
| How do I create a model? | 14 |
| How do I delete a model? | 14 |
| What are domains? | 14 |
| Example: the Host Model structure | 15 |
| What is a data source? | 16 |
| What are “KnowledgeItems”? | 17 |
| | |
| Data modeling tutorials | 19 |
| Modeling example: The org chart | 19 |
| Defining the org structure types | 20 |
| Populating the model using a script | 27 |
| Connecting “Employees” to Host objects | 29 |
| Visualizing the data | 29 |
| | |
| Appendix: Groovy scripts | 30 |
| createOrg Groovy script | 30 |
| | |
| Appendix: Internal database schema | 35 |
| acl_class Table | 39 |
| acl_entry Table | 39 |
| acl_object_identity Table | 39 |
| acl_sid Table | 40 |
| agent_client_defaults Table | 40 |
| agent_config_binder Table | 40 |
| agent_dc_manager_schedule_ids Table | 40 |
| agent_dc_manager_state Table | 41 |
| agent_manager_state Table | 41 |
| alarm_alarm Table | 42 |
| alarm_annotations Table | 42 |

| | |
|--------------------------------------|----|
| alarm_loc_msg Table | 43 |
| auditing_log Table | 43 |
| baseline_config Table | 43 |
| baseline_config_properties Table | 44 |
| baseline_engine_profile Table | 44 |
| baseline_observation_profile Table | 44 |
| cartridge_cartridge_relation Table | 45 |
| cartridge_components Table | 45 |
| cartridge_installed_cartridges Table | 46 |
| cartridge_items Table | 46 |
| credential_data Table | 47 |
| credential_lockbox Table | 47 |
| credential_mapping Table | 48 |
| credential_mapping_entry Table | 48 |
| credential_order Table | 48 |
| credential_policy Table | 49 |
| current_version Table | 49 |
| database_instance_id Table | 49 |
| database_version Table | 50 |
| derivation_calculation Table | 50 |
| derivation_complex_definition Table | 50 |
| derivation_definition Table | 51 |
| fgl4_migration_agent Table | 51 |
| fgl4_migration_data_span Table | 51 |
| fgl4_migration_dcm Table | 51 |
| fgl4_migration_host Table | 51 |
| fgl4_migration_host_mapping Table | 51 |
| fgl4_migration_log Table | 52 |
| fgl4_migration_server Table | 52 |
| incident_affected_objects Table | 52 |
| incident_incident Table | 52 |
| incident_linked_alarms Table | 53 |
| incident_problem_ticket Table | 53 |
| incident_problem_tickets Table | 53 |
| licensing_licenses Table | 53 |
| mgmt_object_size Table | 54 |
| mgmt_observation_size Table | 54 |
| mgmt_timeslice Table | 55 |
| mgmt_timeslice_data_avail Table | 55 |
| model_association Table | 56 |
| model_property_formula Table | 56 |
| model_query_criteria Table | 56 |
| obs_binary_* Tables | 57 |
| obs_metric_aggregate_* Tables | 57 |
| obs_metric_scalar_* Tables | 58 |
| obs_string_* Tables | 58 |
| pcm_encoded_data Table | 59 |
| persistable_config_model Table | 59 |

| | |
|--------------------------------------|----|
| persistable_script Table | 60 |
| persistence_column_mapping Table | 60 |
| persistence_db_column Table | 61 |
| persistence_db_schema Table | 61 |
| persistence_db_table Table | 61 |
| persistence_grouping_policy Table | 62 |
| persistence_lifecycle Table | 62 |
| persistence_lifecycle_period Table | 63 |
| persistence_obs_key_purge_age Table | 63 |
| persistence_obs_purge Table | 63 |
| persistence_obs_purge_age Table | 64 |
| persistence_observation_index Table | 64 |
| persistence_operation Table | 64 |
| persistence_retention_policy Table | 65 |
| persistence_rollup_progress Table | 65 |
| persistence_rollup_retry Table | 66 |
| persistence_storage_config_xml Table | 66 |
| persistence_storage_manager Table | 66 |
| persistence_timeslice_table Table | 67 |
| persistence_topobj_purge_age Table | 67 |
| persistence_type_hierarchy Table | 67 |
| registry_performance_calendar Table | 68 |
| registry_registry_value Table | 68 |
| registry_registry_variable Table | 68 |
| report_output Table | 69 |
| report_schedule Table | 69 |
| rule_action_handler Table | 70 |
| rule_action_message Table | 70 |
| rule_action_registry_reference Table | 71 |
| rule_action_variable_reference Table | 71 |
| rule_blackout_schedules Table | 71 |
| rule_effective_schedules Table | 71 |
| rule_expression Table | 72 |
| rule_firing_strategy Table | 72 |
| rule_messages Table | 72 |
| rule_rule Table | 72 |
| rule_sev_to_clear_actn_hdlr Table | 73 |
| rule_sev_to_fire_actn_hdlr Table | 74 |
| rule_severity Table | 74 |
| rule_severity_expression Table | 74 |
| rule_severity_messages Table | 75 |
| schedule_named_schedule Table | 75 |
| script_annt Table | 75 |
| script_annt_attr Table | 76 |
| script_argument Table | 76 |
| script_argument_annt Table | 76 |
| script_argument_annt_attr Table | 77 |
| script_example Table | 77 |

| | |
|------------------------------------|-----------|
| script_return_annt Table | 77 |
| script_return_annt_attr Table | 78 |
| sec_group Table | 78 |
| sec_group_nesting Table | 78 |
| sec_group_role_match Table | 78 |
| sec_grouprole Table | 78 |
| sec_jaas_source Table | 79 |
| sec_object Table | 79 |
| sec_object_mask Table | 80 |
| sec_object_permission Table | 80 |
| sec_object_type Table | 80 |
| sec_permission Table | 81 |
| sec_permission_def Table | 81 |
| sec_policy Table | 81 |
| sec_resource Table | 82 |
| sec_role Table | 82 |
| sec_user_alias Table | 82 |
| sec_user_obj_permission Table | 82 |
| sec_user_res_permission Table | 83 |
| sec_usergroup Table | 83 |
| sec_userrole Table | 83 |
| sec_x_attribute Table | 84 |
| sec_x_attribute_value Table | 84 |
| tagging_service_mapping Table | 84 |
| threshold_bound Table | 84 |
| threshold_config Table | 85 |
| topology_activity_calendar Table | 85 |
| topology_activity_upgrade Table | 86 |
| topology_object Table | 86 |
| topology_object_history Table | 87 |
| topology_property Table | 87 |
| topology_property_annotation Table | 87 |
| topology_property_history Table | 88 |
| topology_property_name Table | 88 |
| topology_property_value Table | 88 |
| topology_service_state Table | 89 |
| topology_type Table | 89 |
| topology_type_annotation Table | 89 |
| topology_type_history Table | 90 |
| upgrade_pending_operations Table | 90 |
| wcf_groups_by_cartridges Table | 90 |
| wcf_resources Table | 91 |
| About Us | 92 |
| Technical support resources | 92 |

The Foglight data model

This topic provides an introduction to models and discusses the Foglight data model.

- [What are models?](#)
- [Who needs to know about models?](#)
- [Where can I see the models?](#)
- [How are models organized?](#)
- [What are domains?](#)
- [What is a data source?](#)
- [What are “KnowledgeItems”?](#)

What are models?

In general, models are abstractions that capture the essence of the objects they are supposed to represent. A good model looks and behaves like the real thing, at least in certain ways. If a model were perfect in every respect, it would be indistinguishable from the real thing. Thus, we could pose questions, submit these in some way to the model, and obtain the same, or almost the same, results as we would by doing those things to the real object. If the object under consideration undergoes a change, the model would have to change accordingly in order to faithfully represent that object.

The data model used in the Management Server is constructed to do just that. The data sent to the Management Server changes with time, not only because the measurements on properties change, but because the objects themselves may come and go. So, a data model for use with the Management Server must be designed to accommodate the creation of objects, by placing them in a well-designed model hierarchy. Objects have relationships among themselves, and a good model accounts for those relationships.

To the Management Server, models are collections of related data objects. The totality of data objects in existence at any one time is referred to as the “data model”.

- Objects are created by transforming the raw data collected by agents (collection models) or when services are created, deleted, or modified (service models).
- Objects have properties, such as lists and metrics (time series values). Properties may be simple values, but often they are other objects. Being objects, they can have properties that are objects, and these objects may have the starting object as a property. Thus, the relationships form a graph, not a tree.

i **NOTE:** Data objects having other data objects as properties can be a source of confusion when you attempt to drill down into an object’s properties using a data browser. It is possible to encounter a loop of related properties that the data viewer (a tree of nodes) unwinds into a seemingly endless chain of repeating properties. When using the data browser, it is seldom productive to go more than five levels deep. After that, it is likely that you are in an unwound loop.

- Data models can be organized into sub-modules, for instance:
 - HostModel represents a collection of agents on a host.
 - Windows_System represents a Windows® System agent.
 - Physical_Disk represents a disk.

- Raw data can be modeled in different ways.
- You can examine the data model's static skeleton (the defined data types and their inter-relationships) in the Schema Browser.
- You can examine the dynamic data objects in the data browser (**Dashboards > Configuration > Data > Management Server > All Data**).

i | IMPORTANT: Attempting to diagram the data model using these views can entail considerable work. Models should be regarded as being internal to the Management Server.

Who needs to know about models?

Knowledge of the data model is beneficial if you are performing one of the tasks listed in the following table.

Table 1. Who needs to know about models?

| | |
|---|---|
| Extending the model | You might want to add an extra metric to the system, extend an existing collection, or monitor a new and currently unsupported application. |
| Using the query language to build a new UI query | <p>You need to understand the data model to construct a path to the data objects of interest.</p> <p>If you want to write a query to return all the instances of an object of a certain type about which the Management Server currently knows, you can locate them in the data model. Some objects are deeply nested; construct the most efficient path to them.</p> |
| Building your own dashboards | <p>If you intend to design and build your own dashboards, you need to know what the existing data model can supply. Because the Management Server is an open-ended data modeler, it is possible with sufficient effort to build a new data collection framework from the ground up.</p> <p>In brief, the process of building a new data collection framework is as follows:</p> <ul style="list-style-type: none"> • If the data is not already available from existing agents, create an agent to collect the data and install it on target systems. One way is to create a formatted script that writes to <code>STDOUT</code>, upload the script, build a script agent, and then deploy and activate the agent. Consult Quest's Professional Services organization. <p>NOTE: You can use any executable on the client system that writes to <code>STDOUT</code>. All you need is a script to launch the application.</p> <ul style="list-style-type: none"> • Configure the transformation layer that creates typed objects to hold the data and define the interrelationships between data types. Tools: XML and Groovy. <p>Design and build the dashboards to present the data in the Management Server using WCF. Tool: the component editors in Definitions. Resources: the <i>Web Component Tutorial</i> and the <i>Web Component Reference</i> pages.</p> |

Modeling process overview

In this guide, modeling refers to the process of creating new in-memory models. This process can include the creation of:

- A static framework of types and their relationships
- New types when they become needed
- Transformations to create the types as a result of data arrival or model changes
- Instances of those types

