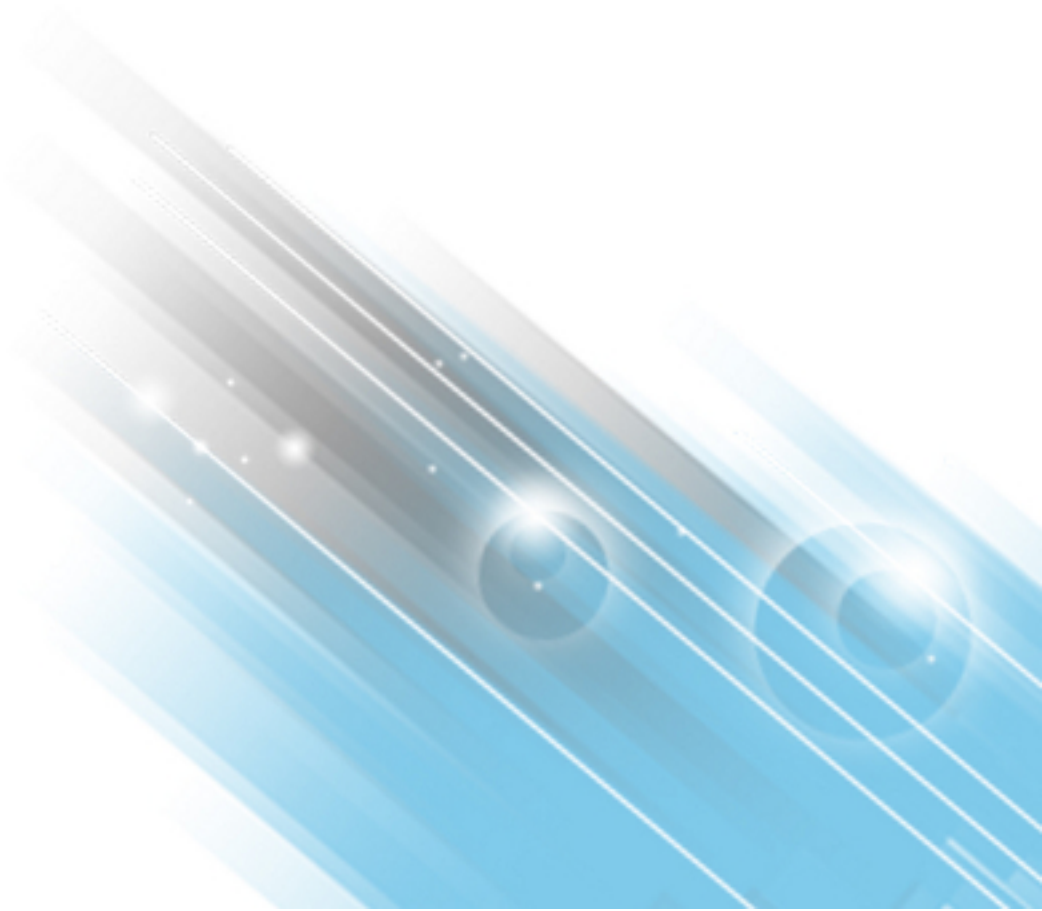
HP Agile Manager

Software Version: 2.40

HP Agile Manager User Guide



Document Release Date: November 2015
Software Release Date: August 2015

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

All third party code is managed by HP Software, and is available upon request.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hp.com.

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to https://softwaresupport.hp.com and click **Register**.

Support

Visit the HP Software Support Online web site at: https://softwaresupport.hp.com

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to: https://softwaresupport.hp.com and click **Register**.

To find more information about access levels, go to: https://softwaresupport.hp.com/web/softwaresupport/access-levels.

HP Software Solutions & Integrations and Best Practices

Visit **HP Software Solutions Now** at https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01702710 to explore how the products in the HP Software catalog work together, exchange information, and solve business needs.

Visit the **Cross Portfolio Best Practices Library** at https://hpln.hp.com/group/best-practices-hpsw to access a wide variety of best practice documents and materials.

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

Getting started

New to Agile Manager? Get started in the following areas:

"Agile Manager life cycles" on page 17 takes through the life cycle of a release or a specific backlog item, with links for more details elsewhere in the Help Center.

Browse through the "Basics and grid actions" on page 23 section for some basic actions commonly used with backlog items.

From each page in Agile Manager, select **Help on This Page** from the **Help** menu on the upper right to take you to the relevant help center topic.

These topics are found throughout the Help Center, such as in the "Product backlog" on page 80, "Release management" on page 107, and "Sprint management" on page 118 sections.

From the **Help** menu in Agile Manager, you can also show or hide product callouts that can help find your way through Agile Manager.

Team member? Watch our Build the product backlog movie.

Planning and tracking releases? Watch our Release planning and tracking movie.

Using the navigation header gives some tips about getting around Agile Manager.

See also:

"What's New – Agile Manager 2.40" on the next page

"Details view" on page 45

"Index of actions" on page 50

"Supported browsers / resolutions" on page 58

"Migrate to Agile Manager" on page 59

Getting started

New to Agile Manager? Get started in the following areas:

"Agile Manager life cycles" on page 17 takes through the life cycle of a release or a specific backlog item, with links for more details elsewhere in the Help Center.

Browse through the "Basics and grid actions" on page 23 section for some basic actions commonly used with backlog items.

From each page in Agile Manager, select **Help on This Page** from the **Help** menu on the upper right to take you to the relevant help center topic.

These topics are found throughout the Help Center, such as in the "Product backlog" on page 80, "Release management" on page 107, and "Sprint management" on page 118 sections.

From the **Help** menu in Agile Manager, you can also show or hide product callouts that can help find your way through Agile Manager.

Team member? Watch our Build the product backlog movie.

Planning and tracking releases? Watch our Release planning and tracking movie.

Using the navigation header gives some tips about getting around Agile Manager.

See also:

"What's New – Agile Manager 2.40" below

"Details view" on page 45

"Index of actions" on page 50

"Supported browsers / resolutions" on page 58

"Migrate to Agile Manager" on page 59

What's New – Agile Manager 2.40

The following features were introduced or enhanced in Agile Manager 2.40:

"Synchronize more with NextGen Synchronizer"

"Define required fields"

"Manage your backlog via API (beta)"

"Use the new forecasting graph to track your release"

"New concurrent licensing model"

"Export and import tasks and acceptance tests"

"Reorder tasks in a backlog item"

"Description field enhancements"

"Add Item dialog box enhancements" on page 16

"Help Center redesign"

Synchronize more with NextGen Synchronizer

NextGen Synchronizer now supports synchronizing past and current releases, including sprints/cycles.

1. Plan your synchronization:
   - Check for releases that already exist in both endpoints, and understand how to synchronize them correctly.
   - Optionally, create a sub-folder in the ALM Releases folder, to use as an alternate root folder.
2. Create a release synchronization link, and define the relevant rules and settings.

For details, see the *Agile Manager Synchronization User Guide*.

This version of NextGen Synchronizer requires a manual upgrade of the Integration Bridge. Future upgrades will take place automatically.

> **Note:** If your Integration Bridge uses a certificate that is not signed by a well known Certificate Authority to communicate with Agile Manager or ALM, you need to reinstall the certificate after upgrading the bridge. For details, see Connections using a certificate that is not signed by a well known Certificate Authority in the *Agile Manager Synchronization User Guide*.

This version also includes the following abilities:

- Synchronize Agile Manager with ALM version 12.21 and 12.50.
- Connect to ALM using LDAP or SiteMinder single sign-on (SSO). For details on SSO connections, see the *Agile Manager Synchronization User Guide*.
- Install the Integration Bridge on Linux or Windows systems.
- On Windows, set up a non-admin user to run the Integration Bridge and Endpoint Credentials Manager.
- On a clustered system, the system administrator can divide the server load by designating nodes for Agile Manager user activity and nodes for synchronization processes. (Agile Manager Administration Site: **Servers > Application**, select **Disable NextGen Synchronizer processes**. For details, see the *Agile Manager Installation and Administration Guide*.)

Define required fields

Site administrators can now define both custom and out-of-the-box system fields as **required** for new items.

Required fields are indicated with a red asterisk (*) in the **Add Item** dialog box.

Users must provide values for required fields when:

- Adding items using the **Add Item** dialog box
- Importing backlog items
- Converting a user story to a feature

To enable this feature, we've changed the **Custom Fields** configuration tab name to **Fields**, and now list both custom and system fields there.

For more details on defining and enforcing required fields, see "Define required fields" on page 163.

Manage your backlog via API (beta)

In addition to the **timesheet** resource, the Agile Manager API now supports the following **general** resources, which developers can use to get lists of Agile Manager items, as well as create, update, and delete individual items via API:

- workspaces
- releases (now also including attachments)
- teams (new!)
- applications (new!)
- backlog_items, including attachments (new!)

  The **backlog_items** resource includes two subtypes: **user_story** and **defect**. Each subtype supports different fields and parameters.

  > **Note:** To support the new backlog_items resource, we added the new **API ID** field to user stories and defects. Use this unique backlog item ID in the API to reference user stories or defects.

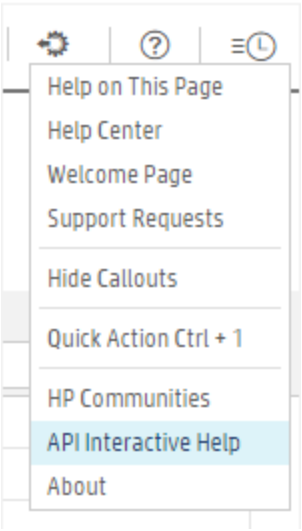
  To view the full list of parameters for each subtype, including any custom fields added to your site, submit a GET operation with a query to filter the response by subtype.

  > For example, use the following query:
  >
  > `"subtype='user_story'"`

API Interactive Help

You can manage Agile Manager items directly from our new API Interactive Help, or copy examples from our interactive help into your code.

Access the Interactive Help from the Agile Manager Help menu (**Help** > **API Interactive Help**), or as a Site Administrator from the **Integrations** > **API** configuration page.



Explore supported entities and operations, or enter a client ID and secret to submit requests to Agile Manager.

If you are new to the Agile Manager API, a site manager must first register your application for integration and provide you with a client ID and secret.

Use the new forecasting graph to track your release

Our new **Release Forecast** graph helps you track and your release based on activity in previous sprints.

The graph displays the forecasted progress until all release content is completed, as well as the release progress so far, for the selected release and team(s).

It highlights any over-planning or under-planning, and when your team will complete release development if you continue at the current pace.

> **Note:** When working in Internet Explorer, the Release Forecast graph is supported only in versions 10 and higher.

For more details, see "Release Forecast" on page 79.

New concurrent licensing model

Agile Manager now supports concurrent licenses, allowing access to a maximum number of simultaneous users.

Administrators can view the following types of information:

For ...	See ...
Concurrent license usage	Our new **Licenses** configuration page (**Site** > **Licenses**). It displays analytic tips based on actual license usage, and a graph showing: • The maximum number of available concurrent licenses; • Peak license consumption points; • The average consumption rate over time. **Note:** This page is available only on systems with concurrent licenses installed.
License usage data	Top-right corner of the **Site** > **Users** configuration page.
Current login status of individual users	**Workspace >Users** and **Site > Users** configuration pages, in the new **Logged In** column.

System administrators can install newly purchased concurrent licenses in the On Premise System Administration site (**Configuration** > **Licenses**).

Caution:

- On one system, you can use Named or Concurrent licenses, but not both.
- Instant On and Evaluation named licenses can be overridden by purchased concurrent licenses, and vice versa.

For details, see the *Agile Manager Installation and Administration Guide*.

Export and import tasks and acceptance tests

Export: You can now include tasks and acceptance tests when exporting backlog items to Excel from the Release Backlog, Product Backlog, and Defect Management.

Import: When you import backlog items from an Excel file, you can now include tasks and acceptance tests.

Create a **Tasks** sheet for defects and user stories, and an **Acceptance Tests** sheet for user stories.

Associate tasks and acceptance tests with backlog items, by adding an **ID** column to the Backlog Items sheet and a **Backlog Item ID** column to the **Tasks** or **Acceptance Tests** sheets.

For assistance, use the import template files available from the Agile Manager Help.

Reorder tasks in a backlog item

Now you can drag and drop tasks in a user story or defect to reorder them, for example, in the order they should be performed.

Drag and drop tasks to reorder them in the **Details** page, the **Sprint Backlog** grid, or the **Task Board**. The task order is retained throughout Agile Manager.

Description field enhancements

Description field changes included in history

When you "View change history", you can now view changes that were made to the Description field.

> **Note:** The history will include only changes performed on Agile Manager 2.40 or later.

Larger edit boxes for Description fields

In the "Details view" on page 45, the editing box for the backlog item **Description** field is now larger and easier to use.

Click anywhere in the Description text to open the edit box. Drag edges and corners of the edit box to re-size it, making it as large or small as you need.

Drag icon for re-sizable Description edit boxes

We added a drag-to-expand icon in the lower-right corner of the **Description** edit box, under the **Cancel** button, to show you exactly where to click to re-size the edit box.

You can find these re-sizable **Description** edit boxes on any "Details view" on page 45 page, whether for a theme, feature, user story, defect, or release.

Increased character limit for tasks and acceptance tests

Descriptions of tasks and acceptance tests now support up to 1000 characters.

Edit tasks in the **Details** view, the **Sprint Backlog** tab, and the **Task Board**.

Add Item dialog box enhancements

Themes filtered by selected application

When you add a new Feature item, select an application.

The list of themes is now limited to those relevant to the selected application.

Attachment field included in Add Item dialog in Internet Explorer

The **Attachment** field is now included in the New Item also when you work in Internet Explorer.

Browse to select an attachment, or drag an attachment into the field.

Help Center redesign

We redesigned our online Help Center with fresh colors and web-friendly, top-navigational layout.

In Agile Manager, select **Help** > **Help on This Page** for details about the page you're looking at.

You can also access our Help Center online, outside of Agile Manager: http://agmhelp.saas.hp.com.

> **Note:** The online version of the Agile Manager Help Center is relevant for the SaaS-based version of Agile Manager. It may describe features that are not available in the on-premise version.

Use the drop down menus at the top to navigate to different areas of the Help Center, and links on the left to navigate to topics in the same area.

Search using filters for the main area of Agile Manager only, or one of our synchronization tools.

For details about earlier releases of Agile Manager, see our **Release Notes** user forum:

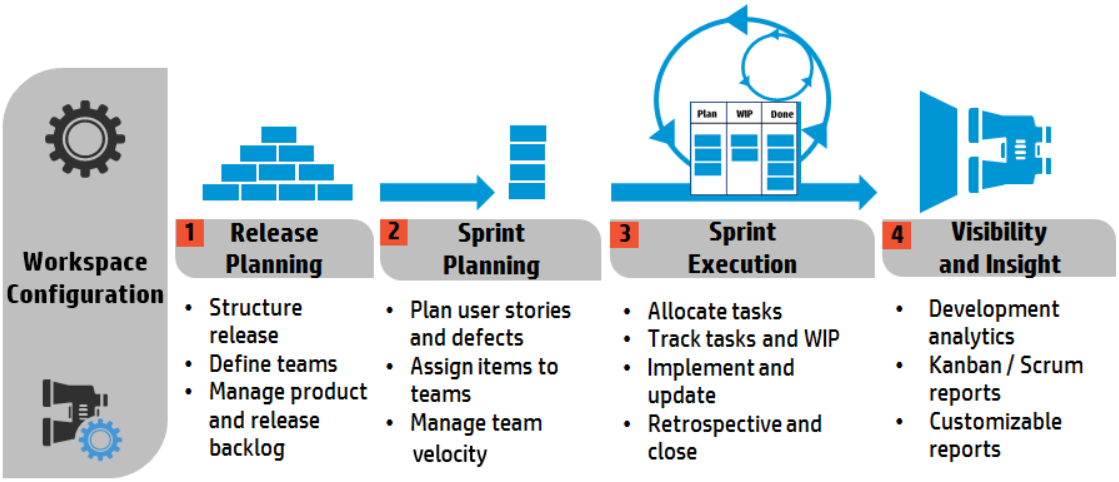
Receive email notifications	Log in via HP Passport, and then select **Options** > **Receive email notifications**.
Subscribe to an RSS feed	Click **Subscribe** at the top of the page. (Your browser must have a built-in RSS feed reader to do this.)

Agile Manager life cycles

This section describes the major processes and concepts in HP Agile Manager.

Release life cycle

The release life cycle in Agile Manager can include the stages described below.



Note: Some steps can be carried out only by Administrator users.

Configuration (Administrators only)

Define workspace settings

- Define the site users assigned to the workspace.
- Configure general workspace parameters that apply to all releases.
- List the applications being developed in your workspace.
- Take advantage of the embedded ALI technology and monitor code development.

  Define build servers and source code repositories to enable data synchronization with Agile Manager.

For more details, see "Configure workspace settings" on page 169 and "Configure ALI integrations" on page 186.

Where: Configuration > Workspace

Release Planning

Structure the release and define teams (Administrators only)

Create a release, and define the teams and team members that will be active in the release.

For details, see "Configure releases" on page 177.

Where: Configuration > Release

Manage product and release backlog

- Define your product themes and features.
- Create a product backlog of user stories and associate them with features and themes.
- From the product backlog, assign user stories and defects to a release.
- When planning a release, pay attention to the release capacity.
- The sum of story points you assign to a team should roughly match the release capacity.
- The release buckets display the amount of story points assigned compared to the overall release capacity.

Where: Product Backlog

Sprint Planning

Plan user stories and defects

From the release backlog, assign user stories and defects to sprints.

Where: Release Management > Release Backlog / Planning Board

Assign items to teams

From the sprint backlog, assign items to teams.

Where: Release Management > Sprint Backlog

Manage team velocity

When planning a sprint, pay attention to the team velocity.

The sum of story points you assign to a team should roughly match the team's sprint velocity.

The team buckets display the amount of story points assigned compared to the overall team velocity.

Where: Release Management > Sprint Backlog

Sprint Execution

Allocate tasks to team members

For each item in the sprint backlog, define the tasks that are necessary to implement the user story or defect.

The tasks can be allocated to the same member as the backlog item, or to other members.

Where: Release Management > Sprint Backlog / Task Board

Track tasks and WIP

When the sprint begins, use the task board to track the progress of your tasks.

Where: Release Management > Task Board

If you using Kanban methodology, use the storyboard to manage the development life cycle of user stories and defects.

Where: Release Management > Storyboard

Implement and update

Develop your product as planned, and update item statuses as you move them towards completion.

Where: Release Management > Task Board / Storyboard

Log defects at any stage of the product development.

Defects are added to the release or sprint backlog in which you report them, and to the overall product backlog.

Where: Defect Management, Release Management > Release Backlog, Sprint Backlog

Retrospective and close the sprint

- Write up a sprint retrospective and create action items.
- Inspect the backlog items that were not completed in the sprint.
- Carry over open backlog items to future sprints.

Where: Release Management > Sprint Closure

Visibility and Insight

Development analytics

Use ALI analytics to continuously monitor the progress of the release development, builds, and source code changes.

Analyze specific features or themes, or work completed by specific teams.

Where: Release Management > ALI Summary; Builds; Source Code

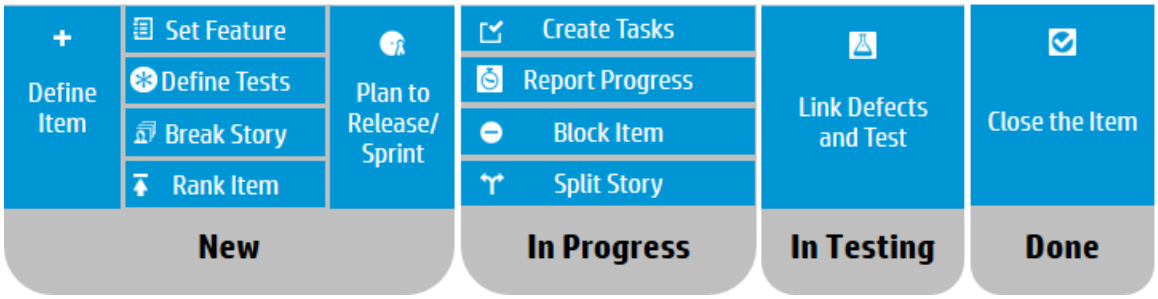
Kanban / Scrum reports and other customizable reports

Use predefined or customized dashboards to track applications, releases, sprints and defects.

Where: Dashboard

Backlog item life cycle

Working with backlog items – user stories and defects – in HP Agile Manager can include the following main steps.



For additional details on each step, see "Sprint management" on page 118 or search the help for other relevant topics.

New

Define a backlog item

1. Define a user story or defect in the product, release, or sprint backlog. All items you create are added to the product backlog, and to the release or sprint backlog in which you created them.
2. Make sure to estimate story points for the backlog item. This enables you to plan releases and sprints according to their capacity.

Assign the backlog item to a feature

Associate the user story or defect with a feature. This enables you to track the development progress on a feature level.

Define acceptance tests

For user stories, define the criteria that must be met in order for the user story to be ready for delivery.

Depending on your project configuration, until all the acceptance tests have passed, a user story cannot be marked as complete.

Break user story

If you defined a general or broad user story, you can break the user story into parts. You can then plan to work on the parts independently in different sprints.

Rank the item

Arrange the backlog by rank so that you can plan the highest ranking items to the next release or sprint backlog.

Plan to release and sprint

- At the release planning stage, drag product backlog items to the release bucket.
- At the sprint planning stage, drag release backlog items to the sprint and team buckets. Alternatively, use the planning board to dynamically move backlog items to sprints and teams.

In Progress

Create tasks

Before you can begin work on a sprint backlog, you need to break the backlog items into tasks.

Tasks describe the actual activities that must be accomplished in order for the user story or defect to be completed. Tasks are measured in hours, as opposed to backlog items that are estimated in story points.

Block the item

If something is preventing you from working on a user story or defect, you can mark the item as blocked. All users can view the reason the item is blocked.

Split user story

At the end of a sprint, if only part of the tasks of a user story were completed, you can split the user story, and transfer the remaining effort to the new user story. This enables you to report the work done on the completed tasks as part of the closed sprint.

Report progress

Progress on a backlog item is reflected in the **statuses** of the backlog item and its tasks, and in the **actual effort** invested and remaining in its tasks.

Statuses	The following rules apply to the statuses of the backlog item and its tasks: • You can manually change the status of a backlog item, or set it to done, at any stage. • Setting a backlog item to done automatically marks all its tasks as completed. • If a new or existing task is then marked in progress, the backlog item is also marked in progress. • If acceptance tests are defined for the user story, all acceptance tests must pass before the user story can be set to done.
Actual Effort	The actual effort of a backlog item is the sum of the time invested and remaining in all its tasks. Compare the actual effort to the planned effort to analyze the accuracy of your estimations.

In Testing - create linkage

- Link the user story or defect to other backlog items.
- Report a defect related to a backlog item.

View the linked items, and navigate between them, in the item's details view.

Done - close the backlog item

At the end of a sprint, all user stories should be closed.

- Agile Manager prevents you from closing a user story that has remaining open tasks or acceptance tests.
- When you close a user story, Agile Manager also notifies you about any linked defects.

A backlog item is usually closed automatically after all its tasks are completed. If you manually set a backlog item to **Done**, all its tasks are automatically set to **Completed**.

Basics and grid actions

This section describes basic features that will help you make the most of Agile Manager.

- Jump across Agile Manager quickly using your Agile Manager "Quick Actions (Ctrl+1)" below.
- "Watch backlog items" on the next page to receive email notifications about a specific item's status updates.
- "Import backlog items" on page 25 from a file into the product or release backlog to migrate data from other systems, or otherwise first list items elsewhere before managing them in Agile Manager.
- "Export backlog items" on page 32 to use and display Agile Manager in other applications.

  You can export backlog items from the following pages into a file:

  - **Product Backlog** > **Themes**, **Features**, **Backlog**
  - **Release Management** > **Release Backlog**, **Sprint Backlog**, **Task Board**
  - **Defect Management**
- "Rank the backlog" on page 33 to help you plan the highest ranking items to the next release or sprint backlog.
- "Create a group story" on page 35 to manage backlog items in a parent-child hierarchy. The parts you define while breaking the story are children of the group story.

Quick Actions (Ctrl+1)

Use the Quick Action shortcut (**Ctrl+1** or **Help** ⓘ > **Quick Actions**) from anywhere in HP Agile Manager to easily navigate to any page, or to perform almost any action.

To select an action, do one of the following:

- To view a list of the available actions in your current context, click the **Show all** link.
- In the action box, start typing the name of an action. Select an action from the filtered list.

The following table lists some of the available Quick Actions.

Note: Available actions vary depending on where you are in Agile Manager.

What do you want to do?	Quick Action name
General	
Navigate to another page	Go To...
Show or hide all callouts	Show Callouts / Hide Callouts
Display all system messages	Reset 'Don't show this again' selections
Navigate to previously viewed page or item	Go Forward / Go Back
Refresh view	Refresh
Create new items	Add New...
Update selected items	Change... To...
Defect Management	
Open a defect by ID	Defect ID:*<defect number>*
Open defect details	View Defect
Assign defects to yourself	Assign to Me
Assign defects to another user	Assign to User:*<user name>*
Delete defects	Delete Defect(s)
Add defects to watch list	Watch Defects(s)
Dashboard	
Add a widget to the dashboard	Add Widgets
Create a custom graph	Add New Custom Graph

Watch backlog items

Watch backlog items and receive mail notifications.

Select backlog items and click **More Actions** > **Watch**.

Watched backlog items are listed in the **My Watched User Stories** and **My Watched Defects** widgets on the Dashboard. When changes are made to watched items, they are displayed in the widgets in bold.

You also receive mail notifications when watched items undergo certain changes. For details, see "Mail notifications" below below.

> **Tip:** In Defect Management, you can add items to another user's watch list. Click **Add Watch for User**.

Mail notifications

Agile Manager sends mail notifications to users when the following events occur:

Event	Who is notified?
Backlog item is assigned to a user	The **Assigned To** user. **Note:** Workspace Administrators can configure whether a mail is sent to the user also if the user assigns the item to him/herself.
Backlog item status changes	All users watching the backlog item, except the user who made the change.
Backlog item exceeds a storyboard time limit For more details, see "Kanban storyboard" on page 131.	All users watching the backlog item.
ALI detects changes in code that may reflect a problem. For details, see "ALI user notifications" on page 144.	The user who checked in the changes. Relevant only when ALI is configured for the release.

Administrators can select which of the events above will be communicated to users. In the Configuration area, open the **Workspace** > **Settings** page.

For details, see "Workspace notifications" on page 171.

Import backlog items

You can import backlog items from a file into the product or release backlog. User stories and defects are imported from separate files.

Tip:

- If you have access to multiple workspaces, be sure you are importing content into the correct workspace.
- Excel files exported from Agile Manager are not formatted in the same way as the import template files. If you export backlog items and import them elsewhere, first copy the content to an import file.

This topic includes:

Prepare a backlog file

In an Excel or CSV file, list the user stories or defects you want to import.

When importing from Excel, your import can also include tasks and acceptance tests. Add a **Tasks** sheet in a defect or user story import file, and/or an **Acceptance Tests** sheet in a user story import file.

The file must be formatted according to specific guidelines, and some fields are restricted to certain values. For more details, see "Import guidelines" on the next page and "Fields and validations" on page 28 below.

Use template files to assist you in creating compatible backlog files.

Template Files

To access import template files, see the Agile Manager Help Center, accessible from the **Help** menu.

Agile Manager supports Microsoft Office Excel versions 2010 and 2013.

Import the backlog items into Agile Manager

1. Select the area in Agile Manager into which you want to import backlog items:
   - **Product backlog:** Under **Product Backlog** > **Backlog**, select the **Backlog Items View**.
   - **Release backlog:** Select **Release Management** > **Release Backlog**. Make sure the correct release is selected.
2. Click **More Actions** > **Import Backlog**.
3. In the Import Backlog dialog box, specify whether you are importing user stories or defects.

After Agile Manager validates the content of the file, you can import the backlog items.

Import guidelines

Review these guidelines to ensure that the import completes successfully. Many of the guidelines are implemented in the template files.

Import guidelines	
Supported file formats	.xls .xlsx .csv
Multiple Excel worksheets	In Excel files, the data must be in the first worksheet, or a worksheet named **Backlog Items**.
Field names and column headers	The field names must be listed on the first row of the file or worksheet. The order of the fields is not important. The column headers in the file must match the field names in Agile Manager.
Merging imported data	You cannot merge an imported backlog item with an existing backlog item: if you import an item with the same name as an existing backlog item, another backlog item will be created with the same name.
Supported fields	You can include any of the standard user story or defect fields, in addition to any custom fields defined in your site. If you want to import data to custom fields, simply add additional columns for the additional fields.

Import guidelines	
Required fields	Some fields are required, and must be filled in for each user story or defect. For details, see "Fields and validations" below below. **Note:** If your administrator has set additional fields as required for user stories or defects, they are not included in the templates by default. You must add them to the import file as additional columns, using the exact names of the required fields. You may want to mark these extra columns as required by adding a comment to the column header, and save the file as a template for later use.
Supported field values	Some fields may contain specific values only. For details, see "Fields and validations" below below.
Multi-list fields	To specify several values in multiple selection fields, separate the values with a semicolon.
Ranking	Backlog items are ranked in Agile Manager according to their order in the file. If backlog items are already defined in Agile Manager, the imported items are ranked below the existing ones.
Themes, features, and applications	Themes and features can belong to multiple applications. If you add separate items with the same theme and feature name, but different applications, any new application values are added to the theme and feature when you import.

Fields and validations

The following tables list the required user story and defect fields that must be filled in the Excel sheet for each backlog item, and the allowed values for certain fields.

First Sheet (User Stories)

Field * required	Allowed Values
Name *	Free text, maximum 255 characters
Description	Free text
Story Points	Positive integer

Field * required	Allowed Values
Priority	Can include one of the following values: 1-High / 2-Medium / 3-Low
Author*	An existing user's login name.
Theme	An existing or a new theme name. New themes will be added to the backlog. Theme names are case-sensitive. **Note:** You can specify a new theme even if the system administrator defined feature fields as required. Field requirements for themes and features are not enforced when you import backlog items.
Feature	An existing or a new feature name. New features will be added to the backlog. Feature names are case-sensitive. **Note:** • If you specified a theme, you must specify a feature. • You can specify a new feature even if the system administrator defined feature fields as required. Field requirements for themes and features are not enforced when you import backlog items.
Application	An application that is already defined in Agile Manager. **Note:** Depending on your workspace or user settings, this field may be required.
Release	To assign a user story to a release, type the name of a release that is already defined in Agile Manager. **Tip:** To import an entire file to a specific release, import it from the Release Backlog page. In this case, you do not need to fill in the Release field.
Sprint	To assign a user story to a specific sprint, type the name of the sprint. The sprint must be included in the specified release.
Team	To assign a user story to a specific team, type the name of the team. The team must be defined in the specified release and active in the specified sprint.

Field * required	Allowed Values
ID	A user story ID, unique within the import file. Required if importing tasks or acceptance tests. Used only to associate a user story with its tasks and acceptance tests, listed in the **Tasks** and **Acceptance Tests** sheets.

First Sheet (Defects)

Field * required	Allowed Values
Summary *	Free text, maximum 255 characters
Description	Free text
Story Points	Positive integer
Severity *	Can include one of the following values: 1-Critical / 2-High / 3-Medium / 4-Low
Priority	Can include one of the following values: 1-Show Stopper / 2-High / 3-Medium / 4-Low
Defect Status *	Can include one of the following values: New / Open / Fixed / Closed / Propose Close / Deferred / Duplicate / Rejected
Detected By *	An existing user's login name.
Detected on Date *	Must include a valid date. In an Excel file, make sure the Detected on Date cells are formatted as dates.
Theme	An existing or a new theme name. New themes will be added to the backlog. Theme names are case-sensitive.
Feature	An existing or a new feature name. New features will be added to the backlog. Feature names are case-sensitive. **Note:** If you specified a theme, you must specify a feature.

Field * required	Allowed Values
Application	An application that is already defined in Agile Manager. **Note:** Depending on your workspace and user settings, this field may be required.
Release	To assign a defect to a release, type the name of a release that is already defined in Agile Manager. **Tip:** To import an entire file to a specific release, import it from the Release Backlog page. In this case, you do not need to fill in the Release field.
Sprint	To assign a defect to a specific sprint, type the name of the sprint. The sprint must be included in the specified release.
Team	To assign a defect to a specific team, type the name of the team. The team must be defined in the specified release and active in the specified sprint.
ID	A defect ID, unique within the import file. Required if importing tasks. Used only to associate a defect with its tasks, listed in the **Tasks** sheet.

Tasks Sheet

Field * required	Allowed Values
Backlog Item ID *	The ID of the backlog item relevant to this task (from the tab containing the defects or user stories)
Task Name *	Free text, maximum 1000 characters
Task Status	Can include one of the following values: New / In Progress / Completed **Default:** New
Task Assignee	An existing user's login name.
Task Estimated Time (Hours)	Positive integer
Invested Hours	Positive integer

Field * required	Allowed Values
Remaining Hours	Positive integer
Color	To apply a color to the imported task, set the cell's fill color in the cell format.

Acceptance Tests Sheet

Field * required	Allowed Values
Backlog Item ID *	The ID of the backlog item relevant to this acceptance test (from the tab containing the defects or user stories)
Acceptance Tests Description *	Free text, maximum 1000 characters
Acceptance Test Status	Can include one of the following values: Not Started / Passed / Failed **Default:** Not Started

Export backlog items

Export grid pages to Excel or CSV files to use and display Agile Manager data in other applications. Exported data includes any columns and filters displayed on the grid, as well as the **Description** and **Comments** fields.

You can export backlog items from the following pages into a file:

- **Product Backlog** > **Backlog**
- **Release Management** > **Release Backlog**, **Sprint Backlog**, **Task Board**
- **Defect Management**

> **Tips:**
>
> - Tasks and acceptance tests are not exported to CSV files.
> - The Task Board supports exporting only to Excel.
> - The **Rank** column is not exported. However, you can first rank items in Agile Manager, and then export them. Items will be listed in the exported file by rank.

1. Define a filter for the items you want to export. On grid pages, arrange columns you want to include.
2. Do one of the following:

On grid pages	Click **More Actions** > **Export Backlog to Excel** or **Export Backlog to CSV**
On the Task Board	Click **Export Backlog to Excel**

3. If you are exporting backlog items to Excel:

   In the **Export Backlog to Excel** dialog box, select whether you want to include tasks or acceptance tests.

   Tasks and acceptance tests are listed in separate tabs in the exported Excel file.

Rank the backlog

A key factor in planning release and sprint backlogs effectively is arranging the backlog by rank. After items in the backlog are ranked, you can plan the highest ranking items to the next release or sprint backlog.

Click the **Rank** column header to sort the grid by rank. Initially, backlog items are ranked in the order in which they were created. To reverse the sort order, click the **Rank** column header again.

Drag a single item up or down the grid to change its rank, or select multiple items and drag them to a new position. Alternatively, enter a new rank number in the **Rank** field. Other items are automatically shifted up or down the grid and re-ranked accordingly.

To shift items to the top or bottom ranks, select the items, and select **Rank Highest** or **Rank Lowest** in the right-click menu.

Backlog items do not have a fixed rank number: in any grid, the items are listed in their relative positions, and ranked consecutively from **1** to **[no. of items in the grid]**.

Note:

- After you set ranks, the ranking is preserved even if you sort the grid by another field. When you sort again by rank, the ranking is restored.
- You cannot display ranking in a grouped grid.

Rank subsets

After you rank a main backlog, you can effortlessly rank any subset of the main backlog. A subset of the product backlog, for example, can be a release or sprint backlog, or a filtered grid.

The items in the subset retain their relative positions from the main backlog, and are ranked consecutively from **1** to **[no. of items in the grid]**.

Changes you make to ranks in the subset are reflected in the main backlog.

Tip: If you want to determine the position of a backlog item relative to other items in the full backlog, right-click the item, and select **Show Position in Backlog**.

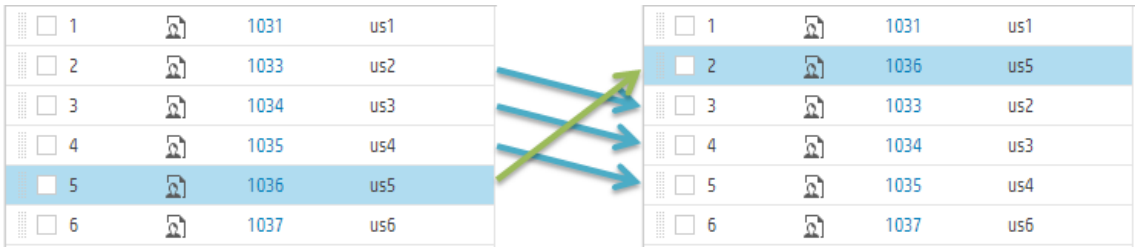
Rank new items

New items that you create are ranked below the lowest ranking backlog item in the product backlog.

Inline ranking

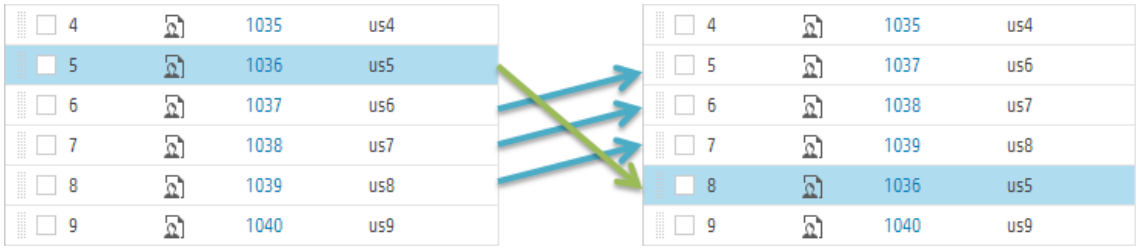
You can edit an item's rank by entering a number in the **Rank** field. The following rules apply:

- **Moving up a rank.** When you move an item to a higher rank *n*, the edited item is moved above the item that was previously ranked *n*. In the example below, **us5** is changed from rank **5** to rank **2**. As a result, items **us2** to **us4** are moved down one rank each.



- **Moving down a rank.** When you move an item to a lower rank *n*, the edited item is moved below the item that was previously ranked *n*. In the example below, **us5** is changed from rank **5** to rank **8**. As a

result, items **us7** to **us5** are moved up one rank each.



Example

To demonstrate some of the auto-ranking behavior, prepare a grid of backlog items, and remove all filters. Then perform the following steps:

Step	Result
1. Sort the grid by rank.	The grid is ranked according to the order in which items were created, or according to ranks you previously set.
2. Drag items up or down the grid.	The items' ranks are set according to their new position in the grid.
3. Define a filter for the grid.	The filtered items are listed in order of their rank, and ranked from **1** to **[no. of items in the grid]**.
4. Drag an item to the second row in the grid. Make a note of the item on the top row.	The item's rank is set according to its new position in the filtered grid.
5. Clear the filter.	The item you moved is ranked immediately below the top ranked item in the filtered grid.

Create a group story

When you break a user story, the original story is converted into a **group story**. The parts you define while breaking a story are children of the group story.

Note: Breaking a user story is different than splitting a user story, which is only done between sprints.

- *Break* a story when you have a large story that you want to divide into smaller pieces. After breaking, the group stories are visible only in the **Group Stories** view on the **Product Backlog** grid.
- *Split* a story when you have not completed all its tasks and acceptance tests by the end of a sprint, and want to push the remaining items to the next sprint. For more details, see "Split a user story" on page 128.

Break a user story

Breaking a user story is useful for dividing a large user story into two or more smaller parts. You can postpone some of the new stories to later sprints or releases, and assign them to other features.

1. In any of the backlog grids, right-click a user story, and click **Break Story**.
2. In the Break Story dialog box, define new user stories that are the parts of the original user story. Assign story points for each new story.
3. Click **Finish**. The new stories are added to the backlog.

Each new child story can also be broken into lower-level parts. You can navigate group stories and their children using the **Group Stories View.**

- The **new stories** inherit the original story's tasks, acceptance tests, and entity links.
- Story points assigned to the **original user** story are removed. The group story's story points are the total story points of its child stories.
- If the **original story** has attachments, the attachments are kept with the group story and are not copied to each child story.

Browse group stories

The Group Stories View enables you to view group stories and drill down to their children. You can also create new stories under a group story.

Open the **Product Backlog** > **Backlog** page, and click the **Group Stories View** link.

- The **Root** level of the Group Stories View displays top level group stories, and user stories that are not children of a group story.
- Click a link in the **Children** column to drill down to the user stories that constitute a group story.
- Use the breadcrumb links to navigate to higher levels of the group story hierarchy.

Grid actions

This section includes:

To show the default display for a specific grid page, click the column selector and select **Restore default display**.

Restoring the default display shows the default columns and removes and filters, grouping, or sorting you defined.

Keyboard selections in grids

When using the column selector and other grid toolbar menus, enter text to jump to and select an item in the list. Press **ENTER** or **SPACE** to select or clear an item.

For example, to add the view details for a selected item, select **More Actions**, and then start typing **View Details**.

The menu will first jump down to the **V** items, and then to the **View Details** item.

Note:

- Keyboard selections work only in top-level menu items. For example, you cannot use enter text to select a specific favorite, as favorites are located within menu folders.
- Keyboard selections can only select items enabled for selection. For example, you cannot select **View Details** if you do not have an item selected in the grid.

Arrange grid columns

Select, reorder, and resize the columns displayed in a grid.

Select columns	Click the **Columns** toolbar button, and mark the columns you want to display in the grid. **Tip:** Start typing a column name to jump to that item in the list. Press **ENTER** or **SPACE** to select or clear an item.

Reorder columns	Position the mouse on a column header and drag it to a new location in the grid.
Resize columns	Position the mouse to the right of a column so that the adjust width icon displays . Drag the icon right or left to increase or decrease the column width.
Hide columns	Click the column action dropdown menu and select **Hide Column** .
Reset columns	In the column selector, select **Restore default display** to restore the columns shown by default

Filter grid items

Filter the items displayed in a grid according to specified conditions. For example, you can create a filter to display only user stories and hide defects, or only user stories higher than a certain rank.

The filter button above the grid is highlighted when a filter is active. Use one of the following methods to filter the grid:

Filter sidebar	1. On the left of the grid, click the **Filter by** button to expand the sidebar. 2. Click the **Add Property** button and select a field to filter by. 3. Select values or type a filter condition. 4. Repeat steps **2-3** to add additional filtering conditions. **Tip:** In a custom list field, if a list value was removed you can still filter by the removed value. Manually type the value in the condition box.
Column filter boxes	In the filter boxes Status located below the column headers, select values or type filter conditions. Where available, click the filter button and select values. **Tip:** • Text fields support the use of an asterisk (*) in the middle of a string. • Numeric fields support the following operators: **<**, **>** , **<=**, **>=**, **<>**, and **""** to display all items with no value • Use a hyphen (**-**) in the **ID** column filter to define a range of backlog items.

Footer	Click links in grid footers to display the relevant data only. For example, in the **Product Backlog** > **Features** grid, if **10% Not Estimated** is displayed in the footer, click the linked text to display only the 10% of the features that have no story point estimation.

Group grid items

Group the items displayed in a grid under the values of a selected field. Use one of the following methods:

Grouping sidebar	1. On the left of the grid, click the **Group by** button to expand the sidebar. 2. Click the **Add Property** button and select a field to group by. 3. To group items with the same value according to the values of another field, click the **+** sign alongside a grouped by field, and select another field to group by.
Column headers	1. Hover over a column header, click the column menu arrow, and select **Group by**. 2. To group items with the same value according to the values of another field, open the column menu of another field and select **Subgroup by**.

Sort grid items

Sort the items displayed in a grid according to the values of a selected field.

1. Hover over a column header, click the column menu arrow, and select **Sort Ascending** or **Sort Descending**. Alternatively, click the column header to sort by the column. Click again to reverse the sort order.
2. To sort items with the same value according to the values of another field, open the column menu of another field, select **Subsort by Column**, and set the sort direction.

Note: If items in the grid are grouped, the sort is applied separately within each group.

Save a favorite view

You can save the current view as a favorite. A favorite saves the following settings: filter, grouping, sort and column layout.

- Organize a grid or dashboard as you like, and click **Favorites > Add to Favorites**.
- To retrieve a saved favorite, click **Favorites**, and select a favorite from the Public or Private folders.
- To create a favorite for use in NextGen Synchronizer, save it as an **Integration** favorite. Fore more details, see the *Agile Manager Synchronization Guide*.

> **Note:** Public favorites are editable by Workspace or Site Administrators only.
>
> For more details, see " Assign users to roles" on page 166.

Export grid items

Export grid pages to Excel or CSV files to use and display Agile Manager data in other applications. Exported data includes any columns and filters displayed on the grid, as well as the **Description** and **Comments** fields.

You can export items from the following pages into a file:

- **Product Backlog** > **Themes**, **Features**, **Backlog**
- **Release Management** > **Release Backlog**, **Sprint Backlog**, **Task Board**
- **Defect Management**

1. Define a filter for the items you want to export and arrange columns you want to include.
2. Click **More Actions** > **Export Backlog to Excel** or **Export Backlog to CSV**.
3. If you are exporting backlog items to Excel:

   In the **Export Backlog to Excel** dialog box, select whether you want to include tasks and acceptance tests.

   Tasks and acceptance tests are listed in separate tabs in the exported Excel file.

Navigate/Browse through grid items

To navigate or browse through the details views for each item in the grid:

1. Click an item ID to open the details view for that item.
2. On the details view, use the arrow buttons to browse to the previous and next items in the filter that was displayed in the grid.

See "Details view" on page 45 for more information about this page.

Edit items

HP Agile Manager offers you several ways to edit backlog items.

Inline editing

You can edit almost any field directly in the grid. Double click the field and type or select the new value.

Edit in form layout

You can display an item's details in a form layout in the "Details view" on page 45, and edit any of its properties or attributes.

1. Open an item's details. Do one of the following:
   - Click the link in the **ID** field, or select **More Actions** > **View Details**.
   - To view item details in a new tab, select **More Actions** > **View Details in a New Tab**. This is useful for comparing multiple backlog items, or for viewing an item's details without leaving the grid.
2. Select any of the editable fields and enter or select new values.
3. Add comments and attachments, define tasks and acceptance tests, and click **Linked Items** to define dependencies with other items.

   Caution: Attachment files can contain dangerous content, and must be downloaded and opened with caution.

   We strongly recommend implementing anti-virus protection for any file storage used when accessing Agile Manager.

4. Make sure to save your changes before leaving the page.

Update multiple items simultaneously

If you need to set the same field value for several items, you can update the items simultaneously. You can set new values for several properties at the same time.

1. In a grid, select the items you want to update, and click **More Actions** > **Update Selected**.

> **Note:** If you select a field whose values are affected by the values another field, you will not be able to also update the other field.
>
> For example, if you select the **Status** field, the **Defect Status** field is disabled, as you cannot manually update values for both of these fields simultaneously.
>
> The values for these fields are automatically updated based on the value of the other field.
>
> For details about this example, see "Manage the defect life cycle" on page 140.

2. In the Update Selected dialog box, click **Add Property**, and select the properties you want to change.
3. Enter or select new values for the properties you selected.

> **Note:** In the **Blocked** field, enter the reason the items are blocked.

4. Click **Update**.

Quick Actions

You can update a single-selection list field using the **Change** Quick Action.

1. In a grid, select one or more items.
2. Press **Ctrl+1** to open the Quick Action dialog box.
3. In the action box, type **Change** to filter the available change actions.
4. Select the change action you want to perform.

For more details, see "Quick Actions (Ctrl+1)" on page 23.

User roles and permissions

Agile Manager users have one or more of the following roles. User roles are assigned per workspace, which means that a single user might have different roles in different workspaces.

> For example, Alex might be a Workspace Administrator in Workspace1, but a Team Member in Workspace2.

User roles are assigned by Site and Workspace Administrators, on the **Site** > **Users** and **Workspace** > **Users** configuration pages. You must be a Site or Workspace Administrator, respectively, to view these pages.

Role name	Description	Configure on..
System Administrator	Has read and write access to the Agile Manager On Premise System Administration site. No default privileges in the Agile Manager application. **Note:** The On Premise System Administration site is accessible to System Administrators only.	The System Administration **Configuration** > **Users** page. For details, see the *Agile Manager Installation and Administration Guide*.
Site Administrator	Has read and write access to **Site** configuration pages, the **Workspace** > **Users** configuration page, and the **Integrations** > **API** configuration page.	The **Site** > **Users** configuration page. For details, see "Define site users" on page 165.
Workspace Administrator	Has read and write access to all application pages and functions, as well as the **Workspace** configuration area.	The **Site** > **Users** or **Workspace** > **Users** configuration pages. For details, see: • "Define site users" on page 165 • "Define workspace users" on page 173

Role name	Description	Configure on..
Team Member	Has read and write access to all application pages and functions, with the following exceptions: • Workspace administrators can set permissions to prevent team members from deleting items created by others. When the **Allow Team Members to delete backlog items created by others** option is cleared, team members can only delete themes, features, and backlog items that they author. By default, this option is selected, and team members can delete any items. Regardless of configuration, Team Members can modify tasks and acceptance tests, regardless of who the author is. • Team members have read-only access to the **Author** field. This field can be modified only by Workspace Administrators. • Team members have read-only access to public favorites on grid pages, to public Dashboard favorites, and to the public Dashboard gallery. Team members can no longer create, update, or delete these items. Team members can also make only the following configuration changes: • Modify team work hours per day and working days in sprint for their team, from the bucket in the **Release Backlog** only. • Modify ALI configurations.	The **Site** > **Users** or **Workspace** > **Users** configuration pages. For details, see: • "Define site users" on page 165 • "Define workspace users" on page 173 • "Configure preferences and notifications" on page 169
Viewer	Has read access only for all backlog items, grid pages, and the **Sprint Closure** page. Can also watch backlog items to receive notifications about status updates. Additionally: • Has read and write access to private favorites on grid pages. • Can create private buckets on the **Defect Management** page. • Has read access on the Dashboard, and read and write access for private Dashboard items and favorites.	The **Site** > **Users** or **Workspace** > **Users** configuration pages. For details, see: • "Define site users" on page 165 • "Define workspace users" on page 173

Role name	Description	Configure on..
Integration Administrator	Has read and write access to the **Integrations** > **Synchronizer** configuration page, for configuring NextGen Synchronizer.	The **Site** > **Users** or **Workspace** > **Users** configuration pages. For details, see: • "Define site users" on page 165 • "Define workspace users" on page 173 • The *Agile Manager Synchronization Guide* • "API integration" on page 225
Integration Bridge	Manages communication between Agile Manager and the NextGen SynchronizerIntegration Bridge. **Note:** For security purposes, this user should have no other roles.	The **Site** > **Users** configuration page. For details, see "Define site users" on page 165 and the *Agile Manager Synchronization Guide*.

Details view

Click the ID of an item, or click **View Details**.

View and edit details of an item in a form layout. Some fields and properties are available only in the details view.

The details view enables you to perform the following main functions:

View and edit fields

- To check if a field is editable, hover over the field value. On editable fields, the value is highlighted, and an edit icon displays alongside the value.
- Hover over the **Theme** or **Feature** values and click the link in the tooltip take you directly to the theme or feature.
- Compare the **Planned** and **Actual** effort values to improve your planning accuracy.

  Planned and **Actual** effort values are aggregated from all of the effort values defined for the item's tasks. **Planned** effort is the sum of all estimated hours, **and** Actual effort is the sum of all invested and remaining hours.
- User stories and defects have a read only **API ID** field. Applications that integrate with Agile Manager using the API use this unique backlog item ID to reference user stories or defects.

Add properties

Property	Description
Comments	Use comments to discuss an item. Click **Add Comment**. **Note:** Comments cannot be deleted after you save them.

Property	Description
Related Tasks	Before you can begin work on a sprint backlog, you need to break the backlog items into tasks. Tasks describe the actual activities that must be accomplished in order for the user story or defect to be completed. Tasks are measured in hours, as opposed to backlog items that are estimated in story points. Drag tasks up or down to organize them, such as in the order they should be performed. **Note:** Depending on your project settings, a default task may already be created for defects.
Acceptance Tests	For user stories, define the criteria that must be met in order for the user story to be ready for delivery. Depending on your project configuration, until all the acceptance tests have passed, a user story cannot be marked as complete.
Attachments	Upload or download a document related to the current item. **Upload a document:** On the right side of the details view, in the **Attachments** box, paste new images or URLs from the clipboard, or drag and drop files from your file system. **Download a document:** In the **Attachments** box, click an attachment to download it. **Modify attached documents:** Click **Attachments** on the left sidebar to view additional details or modify attachments. **Caution:** Attachment files can contain dangerous content, and must be downloaded and opened with caution. We strongly recommend implementing anti-virus protection for any file storage used when accessing Agile Manager.

Send item by email

Send an item to users by email.

On the left sidebar, click **Send by E-Mail**.

Link backlog items and view linkage

Link your item to defects or user stories. You can view the linkage in an interactive diagram or in a grid view.

On the left sidebar, click **Linked Items**.

Create linkage	On the **Linked Items** page, click **Link to User Stories** or **Link to Defects**. Choose whether you want to select items from a grid or to specify their ID. When linking a user story to other user stories, select the **Trace From** or **Trace To** box to determine the direction of the linkage. • A **Trace From** link indicates a user story that impacts the selected user story. This is represented in the diagram by an incoming arrow. • A **Trace To** link indicates a user story that is impacted by the selected user story. This is represented in the diagram by an outgoing arrow.
Navigate to linked items	• In the diagram view, select a tile, and click the arrow. • In the grid view, click the item ID.
Create a new linked item	Click **Add Linked Item**, and complete the fields as you would when adding any other new user story or defect. The new item is automatically linked to the item you were originally viewing.

Tip: To add a new user story or defect that is already linked to a specific item, select the item in a grid, and then select **More Actions** > **Add Linked Item**.

View change history

Review changes that were made to the item.

1. On the left sidebar, click **History**.
2. Filter the records according to the user who made changes, or a changed field.

View development activity

Review the development activity recorded for the backlog item.

On the right sidebar	**Development Metrics** displays the following details for the backlog item: • Lines of code changed • Unit test success rates • Code coverage **Note:** Available metrics depend on ALI and release configuration, and the implementation of the specific backlog item.
On the left sidebar	1. Click **Development Activity**. 2. **Source of metrics.** By default, the build configuration specified as the **Default Source of Metrics** is used to produce development data about the backlog item. To change the source of metrics for the backlog item, select another build configuration from the drop down list. **Data aggregation:** Metrics for parent builds display aggregated data of their downstream builds. 3. Select a view to analyze focused development data about the backlog item: **Commits Change Log** — A list of change sets linked to the backlog item. Open a file, or compare it with a previous version. **Unit Tests** — Information about the unit tests run on code connected to the backlog item. **Code Coverage** — Information about the amount of the backlog item's code that is covered by unit tests. **Active Developers** — Information about the top developers who contributed to the development of the backlog item.

View child items (theme or feature only)

If you are viewing details for a theme or feature, view a grid of child items. On the left sidebar:

If you are viewing a **theme**. . .	Click **Features**.
If you are viewing a **feature**. . .	Click **Backlog Items**.

To view or retrieve archived items, in the grid of child items, select **More Items** > **Display Archive**.

Note: You must be an administrator to retrieve items from the archive. For details, see "Archive

backlog items" on page 97.

Browse through grid items

If you accessed the details view from a grid page, such as the Release Backlog or Sprint Backlog grid, use the **Previous** and **Next** arrow buttons in the top right-corner of the page to navigate to the previous or next item in the grid's filter.

The previous or next button is disabled if you are currently viewing the first or last item in the grid.

Index of actions

This topic includes:

Backlog item actions

Action	Description	Availability
Add Acceptance Test	Define the criteria that a user story must meet in order to be considered complete.	Sprint Backlog, Details page

Action	Description	Availability
Add Attachment	Add an attachment to the selected item. Supported file types are defined per system. For details, contact a system administrator. **Caution:** Attachment files can contain dangerous content, and must be downloaded and opened with caution. We strongly recommend implementing anti-virus protection for any file storage used when accessing Agile Manager.	Add New Item dialog box, Details page
Add Comment	Add a comment to the selected item.	Details page
Add Item	Create a new theme, feature, user story, or defect, depending on the grid you are viewing. When creating a new user story or defect on the **Release Backlog**, **Sprint Backlog**, and **Task Board** pages, the new item is included in the selected release or sprint backlog, and added to the product backlog.	General
Add Linked Item	Create a user story or defect that is linked to the selected backlog item. Based on your settings, a backlog item cannot be closed while it has open items linked to it.	Product Backlog > Backlog, Release Backlog, Sprint Backlog pages
Add Task	Define the actions necessary for fulfilling a user story or defect.	Sprint Backlog, Task Board, Details page

Action	Description	Availability
Block Item / Unblock	Indicate that a backlog item should not be worked on, and specify the reason. A blocked icon is displayed in the grid alongside the item, on the Details page, and in the **Blocked Items** Dashboard widget. Hover over the icon to view the reason. **Note:** Technically, you can assign blocked items to release and sprint backlogs, and change their statuses. However, the best practice is to avoid working on items while they are marked as blocked. When work on the backlog item may resume, unblock the item.	General
Break Story	Break a large user story into two or more smaller stories. The original story is converted into a group story that contains the parts you define. • Each of the new stories inherits the original story's tasks, acceptance tests, and entity links. • Story points assigned to the original user story are removed. The group story's story points are the sum of the story points assigned to the new parts. Use the **Group Story View** on the **Product Backlog** > **Backlog** page to view the group story and drill down to its children. **Note:** You can continue to break the new user stories into lower-level parts. These are all maintained under the same hierarchy as the original user story. **See also:** • "Split Story" on page 54 • "Convert to Feature" below	Product Backlog > Backlog, Release Backlog, Sprint Backlog pages, Task Board
Convert to Feature	If you have a large user story, you can convert it to a feature. You are given the option of keeping the original user story and associating it with the feature. **See also:** • "Break Story" above	Product Backlog, Release Backlog

Action	Description	Availability
Export Backlog to Excel / Export Backlog to CSV	Export a grid to an Excel or CSV file. Grid data is exported as displayed, based on the current filter and selected columns. **Tip:** When you export to Excel, you can include related tasks and acceptance tests.	All Backlog pages, Task Board, Defect Management
Import Backlog	Import backlog items from an Excel or CSV file. **Tip:** When you import from Excel, you can include related tasks and acceptance tests.	Product Backlog, Release Backlog
Mark as Done	Set a backlog item to 'Done'. By using this action, all of the backlog item's tasks are set to 'Completed'. Based on your project settings, you may be prevented from closing a backlog item while it has open linked defects or non-passed acceptance tests.	Release Backlog, Sprint Backlog, Task Board
Navigate to Previous or Next Item	If you accessed the details view from a grid page, such as the Release Backlog or Sprint Backlog grid, use the **Previous** and **Next** arrow buttons in the top right-corner of the page to navigate to the previous or next item in the grid's filter.	Details page
Planning	Plan the selected items to a specified release, sprint, and team. You can specify a release together with a sprint or team, or both.	Product Backlog > Backlog, Release Backlog, Sprint Backlog pages, Task Board, Sprint Closure
Rank Highest / Rank Lowest	Send backlog items to the top or bottom ranks in the backlog.	All Backlog pages
Send by E-Mail	Send items to other project users via email. You can select and send multiple items at the same time.	General
Set Feature	Associate a backlog item with a feature.	Product Backlog > Backlog, Release Backlog, Defect Management

Action	Description	Availability
Show in Backlog Items View	Display the selected user story in the Backlog Items View.	Group Stories view
Show in Group Story	Display the selected user story under its group story in the Group Stories View.	Backlog Items view
Show Position in Backlog	Locate the item in the current backlog. This action clears the filter and highlights the selected item.	All Backlog pages
Split Story	At the end of a sprint, all user stories should be closed. If you invested time in a user story, but not all its tasks and acceptance tests are completed, you can split the story. Remaining effort and non-passed acceptance tests are moved under a new user story, which you can assign to a future sprint. Right-click a user story, and select **Split Story**. **See also:** • "Break Story" on page 52	Sprint Backlog, Task Board, Sprint Closure
Update Selected	Update properties of multiple backlog items at once.	All Backlog pages, Defect Management
View Details / View Details in New Tab	Click an item's ID to view an item's details in a form layout on the Details page. • Right-click and select **View Details in New Tab** to view the details in a separate tab. • Hover over a backlog item to display a tooltip with some basic information and links to the item's feature and theme. Some properties are available only in the Details page and cannot be viewed from the grid.	General

Dashboard actions

Action	Description
Add Widgets	Select widgets from the widget gallery to add to the dashboard.

Action	Description
Change Layout	Set the column configuration of the dashboard.
Clear Dashboard	Remove all the widgets from the current dashboard.
Configure Settings	Tweak the basic settings of an existing graph. Hover over the title banner of a graph, click the down arrow, and select **Configure Settings**. • To return to the graph, click **Save** or **Cancel**. • To configure more settings, such as time frame and filter, click **Advanced Settings**.
Create Custom Graph	Use a wizard to design a graph on any data and time frame you choose. Select **Add Widgets** > **Create Custom Graph**

Task actions

Action	Description
Add Task	Define the actions necessary for fulfilling a user story or defect.
Edit	Edit task details.
Increase remaining / Decrease remaining	Report the time spent on a task, or add to the time planned for a task. Click the **Remaining work** icon, and click the **Decrease remaining** or **Increase remaining** button.
Mark as Completed	Set a task as completed. After you mark a task as completed, the task's remaining hours are automatically transferred to the invested hours.

Sprint closure actions

Action	Description
Add Action Item	Create a new action item.
Convert to action items	After listing the things that worked well or things that can be improved in the sprint, convert them to action items.
Create User Story	Convert an action item to a user story.

Action	Description
The following actions offer you ways to clean up backlog items that remain open at the end of a sprint. ***Click the Open Items tab « located on the right of the Sprint Closure page.***	
Planning	Plan the selected items to a specified release, sprint, and team. You can specify a release together with a sprint or team, or both.
Roll to Next Sprint	Assign the selected backlog items to the next sprint.
Split Story	Move open tasks and acceptance under a new user story. You assign the new story to a future sprint. The completed tasks and passed acceptance tests remain under the original user story. **Note:** Story points are included in the velocity of the sprint in which the user story is closed. Therefore, when you split a user story, the original story's story points are set to zero.

Defect management actions

Action	Description
Add Bucket	Create a new bucket.
Add Item	Create a new defect.
Add to Bucket	Assign the selected defects to a bucket.
Add to My Watch List / Remove from My Watch List **Add Watch for User**	Add defects to your or another user's watch list. Defects in your watch list are displayed in the My Watched Defects widget on the Dashboard.
Assign To	Appoint an owner for the selected defects.
Planning	Plan the selected items to a specified release, sprint, and team.
Remove from All Buckets	Remove the selected defects from all buckets.

Favorite actions

Action	Description
Add to Favorites	Add the current view to the list of favorites. This enables you to reload the view later.
Delete Favorite	Delete the current favorite.
Rename Favorite	Rename the current favorite.
Save Favorite	If a favorite view is loaded and you have made changes to the settings, save the changes to the favorite view.

Drag and drop actions

Action	Description	Availability
Drag and Drop Planning	Drag backlog items to release or sprint buckets.	Product Backlog, Release Backlog
Drag and Drop Ranking	Change backlog item ranks by dragging them up or down the grid.	Backlog Pages
Drag and Drop to Buckets	Drag defects to buckets.	Defect Management
Drag and Drop Attachments	Add attachments by dragging files or URLs to the Attachments field.	Details Page, Add New Item dialog box

Grid actions

Action	Description
Columns	Select the columns to display in the grid.

Action	Description
Sort Ascending / Sort Descending	Click a column header to sort the grid items by the field values of the selected column. Click again to reverse the sort order. **Note:** In a grouped grid, each group is sorted individually.
Subsort by Column	To sort items with the same value according to the values of another field, open the column menu of another field, select **Subsort by Column**, and set the sort direction.

Help actions

Action	Description
Help on This Page	Look up help relevant to the current view.
Help Center	Open the Help Center, offering you links to key information from various sources.
Welcome Page	Display tutorial movies to help you get started with Agile Manager.
Support Requests	**For Portal Users:** Log a request for the HP SaaS support team.
Show Callouts / Hide Callouts	Switch callouts on/off. Callouts provide on-screen assistance in selected areas of Agile Manager.
HP Communities	Open HP Communities and participate in discussions about Agile Manager.

Supported browsers / resolutions

Agile Manager is supported in the following browsers and browser versions:

- Chrome 43 and above
- Firefox 38 and above
- Internet Explorer 9 and above

Note: If you are working in Internet Explorer 9:

- Make sure that the Chrome Frame plug-in is disabled.
- Some functionality is available in versions 10 and above only, such as interactive functionality in Dashboard widgets and velocity graphs.

Screen resolutions

- **Recommended**: 1920x1080
- **Supported**: 1680x1050

Migrate to Agile Manager

Migrating data to Agile Manager consists of the following high-level steps:

1. Export data from your current project management tool.
2. Copy the data to Agile Manager import templates.
3. Use the saved template files to import the data to Agile Manager.

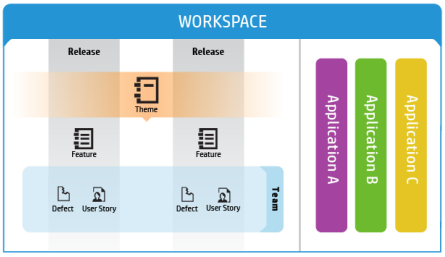
Before migrating, take a few minutes to review the data structure in Agile Manager.

Workspaces

Data in Agile Manager is first divided into Workspaces. Site administrators can create multiple workspaces to represent multiple projects, programs, or products managed on the same Agile Manager site.

Administrators can also assign users to one or more workspaces and define workspace administrators to configure specific settings. Users can only access workspaces that they are assigned to. Data cannot be shared or viewed across workspaces, and users switch back and forth to view data in different workspaces.

Releases and teams are also structured within individual workspaces, and do not cross workspaces.



• **Themes** are the top-level objectives your product must meet, or the product's high-level functional areas. Themes are implemented across releases.

• Themes are broken down into **Features**, which are the areas of a product typically implemented within the scope of a release.

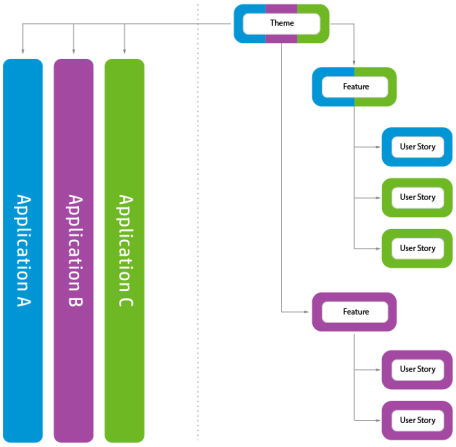
• Features are further broken down into **Backlog Items**, which are the user stories developed and defects fixed by your teams.

• Features and backlog items are developed during the scope of a **Release**, by members of a specific **Team**.

• Themes, features, backlog items, and teams can all be associated with specific **Applications**. Applications are the different components developed by your teams, and can be used to hide data that isn't relevant to you from in grids, graphs, and widgets.

Application inheritance

Associations with applications are inherited from a theme or feature's child items.



When a backlog item is associated with an application, its parent feature and theme is also associated with that application.

Features can consist of several backlog item, implemented in several applications. By the same token, themes can consist of several features, each associated with one or more applications.

Individual backlog items can be associated with only a single application.

Migrating to Agile Manager consists of the following steps:

1. "Filter and export your backlog" on the next page
2. "Download the Agile Manager import templates" on the next page
3. "Compare your current data to the available Agile Manager fields and modify as needed" on the next page
4. "Copy your data to the Agile Manager import templates" on page 63

5. "Prepare your Agile Manager workspace" on page 63
6. "Import data to Agile Manager" on page 64

1. Filter and export your backlog

You'll first need to identify the items you want to migrate to Agile Manager. In your current tool, filter the items you want to migrate, and export them to an Excel file.

It makes most sense to migrate future backlog items only, and not historical information. Agile Manager will not create historical graphs based on migrated data.

> **Caution:** You'll need to migrate user stories and defects into Agile Manager separately. Export them from your current tool separately to ensure that nothing ends up in the wrong place, or duplicated.

Additionally, data is imported to Agile Manager per workspace. If you have data that you want to import to separate Agile Manager workspaces, export them separately to ensure that nothing ends up in the wrong place, or duplicated.

If you have no data hierarchy in your current tool, but you want to retain your currently used tool release trains, migrate your data to the same Agile Manager site and workspace. Use applications within that workspace to separate between types of backlog items.

2. Download the Agile Manager import templates

User stories and defects are imported to Agile Manager using Excel templates, accessed from Agile Manager. Download these templates from the Agile Manager Help Center.

3. Compare your current data to the available Agile Manager fields and modify as needed

When you migrate, you have a variety of options for how to match the data in your current tool with Agile Manager hierarchical elements, depending on how your currently used tool's projects are structured.

Here are some out-of-the-box fields in Agile Manager:

Agile Manager field - user stories	Agile Manager field - defects
Creation date This date is automatically set to the date you import the items.	Detected on Date
-	Defect Status
Description	Description

Agile Manager field - user stories	Agile Manager field - defects
Story Points	Story Points
Priority (User Story) Must be one of the following: **1-High**; **2-Medium**; **3-Low**	Priority (Defect) Must be one of the following: **1-Show Stopper**; **2-High**; **3-Medium**; **4-Low**
Sprint	Sprint
Name	Summary
Author	Detected By
Release	Release
-	Severity Must be one of the following: **1-Critical**; **2-High**; **3-Medium**; **4-Low**

The Agile Manager import templates also provide some additional fields, for **Themes**, **Features**, **Applications**, and **Teams**.

Importing data not supported by default in Agile Manager

Some data exported from your current tool may not be supported in the Agile Manager import template by default. In such cases, you'll need to decide whether you want to import this data, or define it from scratch in Agile Manager.

If you want to import this data, you can do so using one or more of the following options:

Create custom fields	Site administrators can create custom fields in Agile Manager. Custom fields apply to all workspaces in the site. You can create up to 10 fields for user stories, and 10 fields for defects. For details, see "Create custom fields" on page 162. Once you have these custom fields created in Agile Manager, add them as columns to your import templates. Be sure to add only fields that are related to the entity you are importing; add user story fields to the user story template, and defect fields to the defects template. If you have exhausted the maximum number of custom fields available and still have additional data to migrate, migrate data as comments, or as part of the **Description** field.

Migrate additional data as a comment	Add a column to each of the import templates, labeled **Comments**. Copy your additional currently used tool's data to these additional columns, for user stories and defects, respectively. The data is migrated as plain text as a user story or defect comment.
Migrate additional data via the Description field	Copy the additional data to the end of the relevant cells in the **Description** column. This data is migrated as plain text to the user story or defect description.

4. ## Copy your data to the Agile Manager import templates

Copy the data from the file(s) you exported from your current tool to import templates downloaded from Agile Manager. Use the tables in the previous step for recommendations about how to match fields from your current tool and fields in Agile Manager. If you want to associate each item with an application, add that data to the import files.

> **Note:** There are two template files, one for user stories and one for defects. Make sure that you copy the data to the correct template.

5. ## Prepare your Agile Manager workspace

Features and themes are automatically created when you import backlog items, if there are new features and themes listed in the import file.

However, other data, including applications, release data, and any custom fields you require, is not automatically created when you import user stories and defects to Agile Manager. If you define this data in your import files, your Agile Manager site and workspace must be prepared beforehand with the same data.

> **Note:** The names defined in Agile Manager must match those defined in the import files exactly.

The following data must be defined in Agile Manager before you import backlog items associated with this data:

Applications	Defined on the Agile Manager **Workspace** > **Applications** configuration tab. For details, see "Work with applications" on page 82.
Releases	Configured on the Agile Manager **Workspace** > **Releases** configuration tab. For details, see "Configure releases" on page 177.

Teams	Configured on the Agile Manager **Workspace** > **Teams** configuration tab, or for specific releases. For details, see "Manage teams for a release" on page 181.
Sprints	Defined for each release. For details, see "Configure release sprints" on page 180.
Custom fields	Defined on the Agile Manager **Site** > **Custom Fields** configuration tab. For details, see "Create custom fields" on page 162.

6. Import data to Agile Manager

In Agile Manager, you can import data to the **Product Backlog** or **Release Backlog** grids.

- If you are importing all of your data to a single release, import it to the **Release Management** > **Release Backlog** grid.
- If the data you are importing spans multiple releases, import it to the **Product Backlog** > **Backlog** grid.
- If you are importing data to multiple workspaces, do so using separate import files.

To import data, do the following:

a. In the grid header, select **More Actions** > **Import Backlog**.

b. Select the type of items you want to import and click **Let's Begin!**

c. Browse to and select the template file you populated previously. Make sure you select the correct file for the type of item you are importing.

   - Agile Manager validates the data you are importing, and notifies you about the number of items that will be created.
   - Agile Manager also estimates how long the import will take. This is especially useful when importing a very large amount of items.

d. When you're ready to begin, click **Import**. Agile Manager notifies you when the import is complete.

User help

Build your product backlog and assign content to releases and teams.

Track your releases through sprints, or on a Kanban storyboard.

Log defects and track the quality of your releases, builds, and source code.

> **Note:** The Builds and Source Code areas only show content for workspaces configured with ALI for development monitoring.

This section includes the following areas:

Dashboard and analysis tools

Agile Manager provides you with a variety of analysis tools. These include:

Dashboard	The dashboard enables you to display predefined dashboards or design your own. Choose from a widget gallery that includes a wide range of ready-made widgets, or create your own summary or Agile graphs.
Quick widgets	A selection of widgets arranged along the top of Agile Manager pages, enabling you to gain quick insight into the information displayed on the page.
ALI analysis	Using information collected from your build and source code servers, ALI dashboards provide you with intelligent analytics regarding the quality of releases and applications, builds and change sets. **See:** "ALI: Analyze release quality" on page 115, "ALI: Analyze application quality " on page 116, "Development monitoring" on page 142

Note:

- If your widgets are not displayed on the Dashboard, click **Get Started**. To return to the product movies, select **Help** > **Welcome Page**.
- Some advanced widget functionality is limited in Internet Explorer 9.

To use the dashboard, open the **Dashboard** page. To learn more about dashboard widgets, see "Commonly used widgets" on page 68.

What do you want to do?

Select a predefined dashboard

Click **Favorites**, expand the **Public** folder, and select a dashboard for your role – Release Manager, Scrum Master, or Developer. If you use the Storyboard, select the Kanban Dashboard to track the progress of items on the storyboard.

Tip: Some graphs display data based on the current release, sprint, and team. Make sure the correct release, sprint, and team are selected at the top of the page.

Add a graph to a predefined dashboard

1. Make sure the correct dashboard is selected. The dashboard name is displayed alongside the **Favorites** label.
2. Click **Add Widgets**. The Add Dashboard Widgets dialog box opens.
3. Choose a category, or type a string in the search box to find a widget.
4. Select a widget, and click **Add to Dashboard**. The widget is added to the bottom row of the current dashboard.

Save a dashboard

After you have designed a dashboard, you can save it as a favorite. You can then return to the same dashboard in the future. A favorite stores the graphs and their layout on the page.

Select **Favorites** > **Add to Favorites**.

Tip: Favorites in the Public folder can be accessed and modified by all users.

Create a custom graph

Agile Manager provides an array of predefined graphs in different categories. In addition to these, you

can create custom graphs, save them to the widget gallery, and include them in any dashboard.

You can create two types of custom graphs:

Summary Graph	Provides a snapshot view of backlog items (user stories, defects, or both), according to criteria that you specify. For example, you can create a pie chart of user story statuses for a specific sprint, or a column chart of requirements assigned to users, grouped by status. Summary graphs can also display data defined in custom fields. Add them to your custom summary graph by selecting the custom field in the **Filter** or **Display** wizard pages, as needed. • Supported custom field types include: **Date**, **Free Text**, **Single Value List**, **Numeric**, and **Users**. • **Multi Value List** fields are not supported in custom Summary graphs.
Agile Graph	Displays the development of backlog item data over time, according to criteria that you specify. For example, you can track the number of user stories remaining in a team's backlog on any given day during a sprint. Additionally, define the data used for the X- and Y-axis, and how data is grouped. Custom Agile graphs can show custom **Single Value List** and **Numeric** fields defined for user stories and defects, which are enabled for tracking. • To view these custom fields in Agile graphs, edit the field on the **Site** > **Fields** configuration page, and select the **Track this field over time...** option. For details, see "Track custom fields over time" on page 162. • **Date**, **Free Text**, **Multi Value List**, and **Users** fields are not supported in custom Agile graphs.

For more details, see "Create custom fields" on page 162.

To create a custom graph:

1. Click **Add Widgets**. The Add Dashboard Widgets dialog box opens.
2. Click **Create Custom Graph**. The New Custom Graph wizard opens.
3. Select a graph type, and follow the wizard steps. Save your graph to the gallery, in a private category to access it again later, or in a public category to enable others to view it too.

Change graph settings

You can change settings of graphs displayed in the dashboard. The basic settings you can adjust include: Title, Description, Release, Team, and Granularity.

1. Hover over the title banner of a graph, click the down arrow  , and select **Configure Settings**.
2. Change the settings, and click **Save**.

> **Tip:** By selecting the **Context Sensitive** value in the Release, Sprint or Team fields, you can force the graph to display data according to the release, sprint, and team selected at the top of the page.

Commonly used widgets

This topic provides details about commonly used Agile Manager dashboard widgets.

See also: "Release Manager dashboard" on page 113 and "Daily sprint tracking" on page 126

> **Note:** When working in Internet Explorer, interactive functionality in Dashboard widgets and graphs is available in versions 10 and above only.

- "My Stories" on the next page
- "My Defects" on the next page
- "Blocked Items" on page 70
- "Team Workload" on page 78
- "Release Backlog Cumulative Flow Diagram" on page 77
- "Team Velocity" on page 71
- "Group Velocity" on page 71
- "Sprint Control Chart" on page 72
- "Release Control Chart" on page 72
- "Sprint Burn Up" on page 74
- "Sprint Burn Down" on page 76
- "Release Burn Up" on page 73
- "Release Burn Down" on page 75
- "Sprint Planned vs Actual Backlog Items" on page 79
- "Sprint Planned vs Actual Backlog Items (by Story Points)" on page 79
- "Release Planned vs Actual Backlog Items" on page 78
- "Release Planned vs Actual Backlog Items (by Story Points)" on page 78
- "Release Forecast" on page 79

My Stories

Description	Displays user stories assigned to you or created by you, depending on widget configuration. If the widget is configured to display stories assigned to you, it also includes stories with tasks assigned to you, even if the story itself is assigned to another user. This widget does not display user stories with a status of **Done**.
Interactive	

Display by status	Display stories of a specific status by clicking the status name at the top of the widget. Click **User Stories** to show stories of all statuses (except for **Done**).
Display by blocked or watched status	Display only stories that are blocked, or only stories that you are watching, by clicking the / icons at the top of the widget. Click both icons to display only stories that are both blocked and watched by you.
Navigate many items	If you have many stories displayed, page through the widget using the arrow buttons at the bottom.
Modify status	Click the status dropdown on the left of a user story row to modify the story's status.
More actions	Hover over a user story row to display the **More Actions** dropdown on the right. Click the arrow to do one of the following: • Send the user story to another user by email • Block/unblock the user story • Watch/unwatch the user story

Back to top

My Defects

Description	Displays defects assigned to you or created by you, depending on widget configuration. If the widget is configured to display defects assigned to you, it also includes defects with tasks assigned to you, even if the defect itself is assigned to another user.

Interactive	**Display by status**	Display defects of a specific status by clicking the status name at the top of the widget. Click **Defects** to show stories of all statuses. By default, this widget does not display defects with **Closed** statuses. Modify widget settings to add these statuses.
	Display by blocked or watched status	Display only defects that are blocked, or only defects that you are watching, by clicking the / icons at the top of the widget. Click both icons to display only defects that are both blocked and watched by you.
	Navigate many items	If you have many defects displayed, page through the widget using the arrow buttons at the bottom.
	Modify status	Click the status dropdown on the left of a user story row to modify the defect's status.
	More actions	Hover over a defect row to display the **More Actions** dropdown on the right. Click the arrow to do one of the following: • Send the defect to another user by email • Block/unblock the defect • Watch/unwatch the defect

Back to top

Blocked Items

Description	Displays items in the release that are currently blocked.	
Available actions	**Open the item**	Click the item name to open its Details page.
	View all items in a grid	At the bottom of the widget, click **View All...**. All blocked items in the release are displayed in a filtered grid.

Back to top

Team Velocity

Description	For a specific release and team: The number of story points completed by the team in the past and current sprints of the specified release, compared with the average number of story points completed in past sprints of the specified release.
Velocity (columns)	The number of story points completed in a sprint, and within 7 days after the sprint end date.
Average Velocity (line)	Takes into account all past sprints in the release. It does not take into account the story points closed in the current sprint.
Time Scale	Sprints in the release
Frequency	Sprints

Back to top

Group Velocity

Description	For a specific release and team: The number of story points completed by all teams in the past and current sprints of the specified release, compared with the average number of story points completed in past sprints of the current release.
Velocity (columns)	The number of story points in a sprint, and within 7 days after the sprint end date.
Average Velocity (line)	Takes into account all past sprints in the release. It does not take into account the story points closed in the current sprint.
Time Scale (X-Axis)	Sprints in the release
Frequency	Sprints

Back to top

Sprint Control Chart

Description	For a specified release, sprint, and team: The cycle times of backlog items delivered in the sprint. The graph indicates the items' cycle times relative to the average cycle time. Click individual items in the graph to display additional data in a tooltip. **Note:** Chart includes only items modified within the sprint start and end dates.
Days (Y-Axis)	The total number of days in an item's cycle time.
Time Scale (X-Axis)	Work days in the sprint, indicating the date on which the item was marked as **Done**.

Back to top

Release Control Chart

Description	For a specified release and team: The cycle times of backlog items delivered in the release. The graph indicates the items' cycle times relative to the average cycle time. Click individual items in the graph to display additional data in a tooltip. **Note:** Chart includes only items modified within the release start and end dates.
Days (Y-Axis)	The total number of days in an item's cycle time.
Time Scale (X-Axis)	Work days in the sprint, indicating the date on which the item was marked as **Done**.

Back to top

Release Burn Up

Description	Displays the progress towards completing the release plan. The graph compares the number of story points at any point on the release timeline, with the number of planned story points at the same point in time. **Note:** Graph includes only items modified within the release start and end dates.
Story Points	The total number of story points planned for the release. The value can change as story points are added to– or removed from– the release plan. Changes in scope are highlighted by vertical lines in the graph, indicating additional items planned on the specified day.
Done Story Points	The total number of release story points that are marked as 'Done' at the end of any specific day during the release.
Time Scale (X-Axis)	Work days in the release. Each point on the X-Axis represents the end of the indicated day, and Y-Axis values are the values for the end of that day. The origin represents the beginning of the first day of the release.
Frequency	Daily
Interactive	• Click on legend items to add or remove content from the graph. • Hover over spots in the graph to view more data. **Note:** When working in Internet Explorer, interactive widgets are available in versions 10 and above only.

Back to top

Sprint Burn Up

Description	Displays the progress towards completing the sprint plan. The graph compares the number of hours invested so far in the sprint, with the total number of hours assigned to the tasks planned for the sprint. **Note:** Graph includes only items modified within the sprint start and end dates.
Sprint scope	The total number of hours estimated for the tasks at the end of any specific day in the sprint. The value can change as tasks are added to– or removed from– the sprint plan. Changes in scope are highlighted by vertical lines in the graph, indicating additional items planned on the specified day.
Invested effort	The total number of hours invested in sprint tasks at the end of any specific day of the sprint.
Time Scale	Work days in the sprint. Each point on the X-Axis represents the end of the indicated day, and Y-Axis values are the values for the end of that day. The origin represents the beginning of the first day of the sprint.
Frequency	Daily
Interactive	• Click on legend items to add or remove content from the graph. • Hover over spots in the graph to view more data. **Note:** When working in Internet Explorer, interactive widgets are available in versions 10 and above only.

Back to top

Release Burn Down

Description	The remaining open story points for all teams, on any specific day in the release, compared with where the teams should be, based on the expected velocity. **Note:** Graph includes only items modified within the release start and end dates.
Remaining Story Points (curve)	The number of story points associated with all open user stories and defects assigned to the release, by the end of any specific day in the release.
Expected Velocity (diagonal line)	A guide as to the amount of work (open story points) that should remain for all teams, by a specified day in the release. At the start of the first day of the release, the teams' **expected velocity** is the sum of all teams' expected sprint **velocities***, multiplied by the number of sprints. At the end of the last day of the release, the expected velocity (remaining work) is zero. The diagonal line connects these two points. **Note:** If a team's expected velocity is changed during a release, the new expected velocity is adjusted from the beginning of the release timeline. (*) Each team's **expected sprint velocity** can be found in the configuration area, on the team details configuration page, under **Est. Sprint Velocity**. For details, see "Configure team settings" on page 183.
Time Scale	Work days in the release. Each point on the X-Axis represents the end of the indicated day, and Y-Axis values are the values for the end of that day. The origin represents the beginning of the first day of the release.
Frequency	Daily

Interactive	• Click on legend items to add or remove content from the graph. • Hover over spots in the graph to view more data. **Note:** When working in Internet Explorer, interactive widgets are available in versions 10 and above only.

Back to top

Sprint Burn Down

Description	The team's remaining work hours on any specific day in the sprint, compared with where the team should be, based on the team's calculated capacity. **Note:** Graph includes only items modified within the sprint start and end dates.
Remaining Effort (curve)	The number of work hours associated with open tasks in the sprint.
Capacity (diagonal line)	A guide as to the amount of work that should remain for the team by a specified day in the sprint. On the start of the first day of the sprint, the team capacity (remaining work effort in hours) is equal to the sum of each member's capacity for the sprint. Each member's capacity for the sprint is equal to the member's work hours per day, multiplied by the number of work days for the team member in the sprint. This value is found in the team member planning buckets on the right side of the **Sprint Backlog** page (**Release Management** > **Sprint Backlog**). **Note:** If a member's capacity is changed during a sprint, the new team expected capacity is adjusted from the beginning of the sprint timeline.
Time Scale	Work days in the sprint. Each point on the X-Axis represents the end of the indicated day, and Y-Axis values are the values for the end of that day. The origin represents the beginning of the first day of the sprint.
Frequency	Daily

Interactive	• Click on legend items to add or remove content from the graph. • Hover over spots in the graph to view more data. **Note:** When working in Internet Explorer, interactive widgets are available in versions 10 and above only.

Back to top

Release Backlog Cumulative Flow Diagram

Description	Displays the number of backlog items in each status at any specified time in the release. • Looking vertically, you can see the distribution of backlog items across different statuses at any point in the release. • Looking horizontally, you can track the progression of statuses along the release timeline. **Note:** Diagram displays changes made only within the release start and end dates.
Time Scale	Work days in the release. Each point on the X-Axis represents the end of the indicated day, and Y-Axis values are the values for the end of that day. The origin represents the end of the first day of the release.
Frequency	Daily
Interactive	• Click on legend items to add or remove content from the graph. • Hover over spots in the graph to view more data. **Note:** When working in Internet Explorer, interactive widgets are available in versions 10 and above only.

Back to top

Team Workload

Description	The number of work hours remaining for each team member until the end of the sprint.
Remaining Work within Capacity (blue bar)	The number of work hours associated with open tasks in the sprint, and assigned to the team member.
Remaining Free Capacity (white bar)	The amount of free capacity the team member has remaining. At the bottom of the widget, the number of **Hours of unassigned tasks** refers to hours of tasks that are assigned to the sprint, but not assigned to any team member.
Remaining Work—Over Capacity (red bar)	The number of work hours assigned to the member that exceeds capacity.

Back to top

Release Planned vs Actual Backlog Items

Planned	Backlog items that were assigned to sprints before– or within two days of– the beginning of each sprint. Each column aggregates planned items for all the release sprints, and groups them by done and not done items.
Added	Backlog items that were assigned to sprints later than two days from the beginning of each sprint. Each column aggregates added items for all the release sprints, and groups them by done and not done items.

Back to top

Release Planned vs Actual Backlog Items (by Story Points)

Planned	Backlog items that were assigned to sprints before– or within two days of– the beginning of each sprint. Each column aggregates planned items for all the release sprints, and groups them by done and not done items. Backlog items are expressed here in terms of story points.
Added	Backlog items that were assigned to sprints later than two days from the beginning of each sprint. Each column aggregates added items for all the release sprints, and groups them by done and not done items. Backlog items are expressed here in terms of story points.

Back to top

Sprint Planned vs Actual Backlog Items

Planned	Backlog items that were assigned to the sprint before– or within two days of– the beginning of the sprint. Backlog items are grouped by done and not done items.
Added	Backlog items that were assigned to the sprint later than two days from the beginning of the sprint. Backlog items are grouped by done and not done items.

Back to top

Sprint Planned vs Actual Backlog Items (by Story Points)

Planned	Backlog items that were assigned to the sprint before– or within two days of– the beginning of the sprint. Backlog items are grouped by done and not done items. Backlog items are expressed here in terms of story points.
Added	Backlog items that were assigned to the sprint later than two days from the beginning of the sprint. Backlog items are grouped by done and not done items. Backlog items are expressed here in terms of story points.

Back to top

Release Forecast

Description	Displays the forecasted progress until all release content is completed, as well as the release progress so far, for the selected release and team(s). **Note:** When working in Internet Explorer, the Release Forecast graph is supported only in versions 10 and higher.
Release Scope (dark purple line)	The total number of story points estimated for the release's backlog items at the end of any specified day in the release. The value can change as items are added to– or removed from– the release plan. Changes in scope are highlighted by diagonal lines in the graph, indicating additional items planned on the specified day.

Planned Velocity (light purple line)	A guide as to the amount of work that should be completed for all teams, by a specified day in the release. This line is based on the Expected Velocities for all teams in the sprints.
Done (green line)	Backlog items marked as **Done** by the current date.
Forecast (dotted line): • green if release is ahead • red if release is in delay	The forecasted progress until the teams complete the currently planned content, based on the settings configured. This line can change as user stories or defects are added to– or removed from– the sprint plan.
Today (light red vertical line)	Marks the current day in the release.
Planned End (light red vertical line)	Marks the scheduled end of the release.
Time Scale	Work days and sprints in the release. Each point on the X-Axis represents the end of the indicated day or sprint, and Y-Axis values are the values for the end of that day or sprint. The origin represents the beginning of the first day of the release.
Frequency	Daily
Interactive	• Click on legend items to add or remove content from the graph. • Hover over spots in the graph to view more details.

Back to top

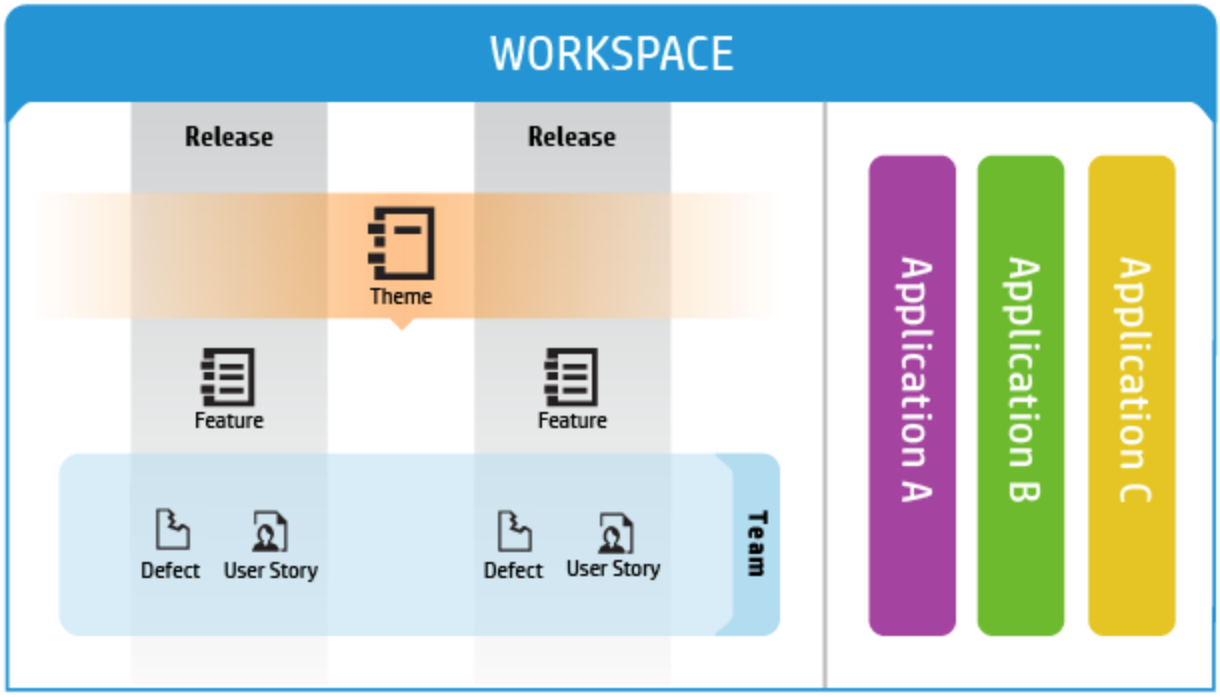
Product backlog

Data in Agile Manager is first divided into **Workspaces**. Site administrators can create multiple workspaces to represent multiple projects, programs, or products managed on the same Agile Manager site.

Administrators can also assign users to one or more workspaces and define workspace administrators to configure specific settings. Users can only access workspaces that they are assigned to. Data cannot be shared or viewed across workspaces, and users switch back and forth to view data in different workspaces.

Releases and **teams** are also structured within individual workspaces, and do not cross workspaces.

The following diagram illustrates how items are organized in a workspace.



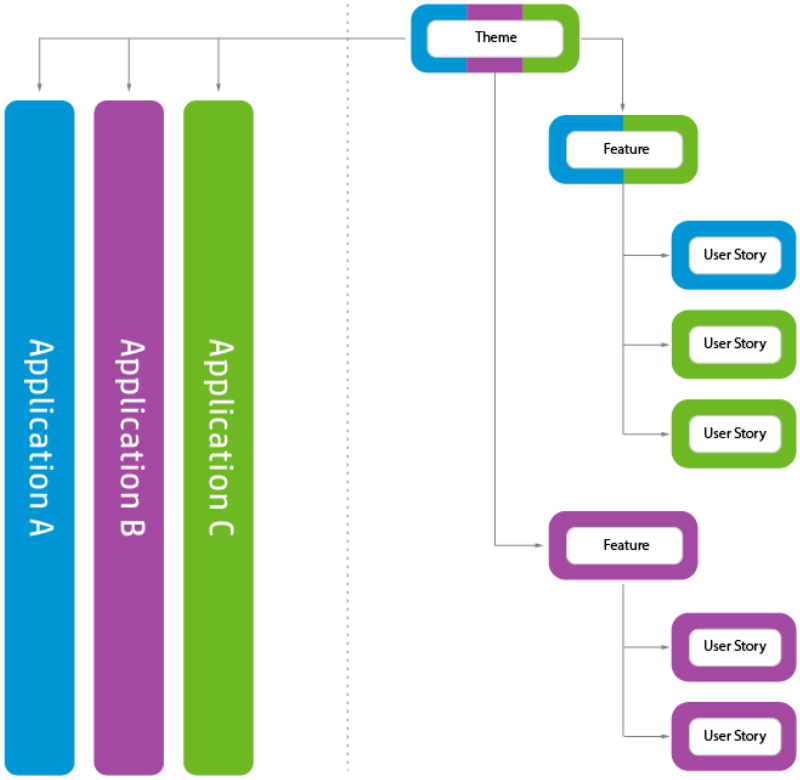
Themes are the top-level objectives your product must meet, or the product's high-level functional areas. Themes are implemented across releases.

- Themes are broken down into **Features**, which are the areas of a product typically implemented within the scope of a release.
- Features are further broken down into **Backlog Items**, which are the user stories developed and defects fixed by your teams.
- Features and backlog items are developed during the scope of a **Release**, by members of a specific **Team**.
- Themes, features, backlog items, and teams can all be associated with specific **Applications**. Applications are the different components developed by your teams, and can be used to hide data that isn't relevant to you from in grids, graphs, and widgets.

This section includes the following:

Application inheritance

The following diagram illustrates how an association with an application is inherited from a theme or feature's child items.



When a backlog item is associated with an application, its parent feature and theme is also associated with that application.

Features can consist of several backlog item, implemented in several applications. By the same token, themes can consist of several features, each associated with one or more applications.

Individual backlog items can be associated with only a single application.

Work with applications

Applications are the separate components developed by the teams in your workspace. Applications can be independent of each other, or composite – designed to be integrated into a single product.

You can associate the following items with applications: **themes**, **features**, **user stories**, **defects**.

Items that do not belong to any application are assigned to the **not set** category. Themes and features can be associated with multiple applications.

Applications trickle up the product backlog tree, and do not trickle down:

- When you associate child items with applications, the parent items are automatically associated with those applications. Group stories are not affected.
- When you associate parent items with applications, the child items are *not* automatically associated with those applications. You can associate the child items with the applications separately.
- If child items are not associated with any application, they and their parent items are assigned to the **not set** category.

Set up applications

1. **Define applications.**
   - In the Configuration area, on the **Workspace** > **Applications** page, create a list of applications.
   - On the **Workspace** > **Users** page, specify the applications users have access to. To require users to specify the application for new items, clear the (not set) value from the Applications field.
   - To require all users to specify the application for new items, select options in **Configuration** > **Settings** > **Set required fields**.

   > **Note:** After providing a user with access to an application, the user must log out of Agile Manager and log in again for the changes to take effect.

2. **ALI: Associate build configurations with applications.** This enables you to filter builds by application.

   On the **Workspace** > **Builds** page, expand a build server, and click a build configuration. Specify an application.

3. **ALI: Associate repository branches with applications.** This enables you to filter code changes by application.

   On the **Workspace** > **Code** page, expand a source code server, and click a branch. Specify an application.

Data hiding

Workspace administrators can specify which applications users have access to. Agile Manager displays only items that are connected to the applications to which the user has access.

- **Administrators:** In the Configuration area, on the **Workspace** > **Users** page, specify the applications users have access to.

  > **Note:** Items that do not belong to any application are assigned to the **not set** category. Determine whether users have access to such items.

- **Users:** The information displayed in Agile Manager is filtered according to the applications to which you were given access.

  You can further filter the information by a specific application: select an application from the **Application** selector, or define a filter on the Applications column.

> **Note:**
>
> - Items can belong to the **not set** category. Such items will be displayed only if you have access to the **not set** category.
> - A feature or theme may be associated with several applications. The feature or theme will be displayed as long as you have access to at least one of its applications.

Analyze applications with ALI analytics

1. **Review overall application health and quality.**

   Use the Application Summary view to gain insight into the quality and health of applications in a release, based on code change, defect, and build information.

   For more details, see "ALI: Analyze application quality " on page 116.

2. **Review build and code change information for applications.**

   Filter build and change set information by application. Open the **Builds** or **Source Code** page, and select an application to filter by.

Track application progress

You can track the progress of an application on the Features and Themes pages, or in a widget on the Dashboard.

Use the following methods to track application progress:

- **Features and Themes pages:** On the Features or Themes page, filter the grid by **Application**. Only the items belonging to the specified application will be listed. Use the progress bars to track the work on the application's features or themes.
- **Dashboard:** On the Dashboard, use one of the following:
  - In the widget gallery, select from a selection of widgets under the **Applications** category.
  - Customize other widgets to display data by application.

Define themes

Define product themes and track their development progress.

This topic includes:

Add themes

1. Select the **Product Backlog** > **Themes** tab.
2. Click **Add Item** and define your product themes.

Rank themes

Ranking themes can assist you prioritize the planning of associated features and backlog items.

1. Click the **Rank** column header to sort the grid by rank. Initially, themes are ranked in the order in which they were created.
2. Drag the item up or down the grid to change its rank. Alternatively, enter a new rank number in the **Rank** field. Other items are automatically shifted up or down the grid and re-ranked accordingly.

Attach a document to a theme

To upload documents related to a theme, right-click a theme and select **View Details**. On the right side of the details view, in the **Attachments** box, paste new images or URLs from the clipboard, or drag and drop files from your file system.

Click **Attachments** on the left sidebar to view additional details or modify attachments.

Tip: In the details view, you can also add comments on the item, and view the item's change history.

Associate a theme with applications

Specify which of your applications a theme relates to.

Select a theme, and click **Set Applications**.

> **Note:**
>
> - A theme automatically inherits the applications that its features and backlog items are associated with. You cannot remove these applications from the theme.
>   Applications inherited from its features and backlog items are listed in the **Applications** column in the Themes grid.
> - Features and backlog items are not automatically associated with a theme's applications.

Drill down to a theme's features

The Features column displays the number of features associated with each theme.

To display the features associated with a theme, click a value in the **Features** column. The Features page opens, and displays the features that belong to the theme you selected.

Track theme progress

You can track the progress of a theme on the Themes page, or in a widget on the Dashboard.

Themes page

- Follow the progress bar for each of the themes. The progress bar represents the amount of the theme's completed story points as a percentage of the theme's total story points.

  Hover over the progress bar for a detailed list of completed and remaining story points, as well the number of user stories and defects.
- To track the progress of a theme in a particular release or application, filter the **Applications** or **Releases** columns. The progress bars aggregate only those backlog items that are assigned to the selected applications or releases.

Dashboard page

Track the development of a theme in a specified release or application. Use one of the following:

- In the widget gallery, under the **Releases** category, select the **Theme Status** widget.
- Select an out-of-the-box release or application widget, or create a custom widget, and configure it to display data by theme.

Define and prioritize features

Define product features and track their development progress.

Features can be prioritized by rank, or by the various Weighted Shortest Job First (WSJF) components: Business Value, the Cost of Delay (CoD) sum, or the overall WSJF score.

The WSJF formula is a well-established method for determining feature priorities. It consists of various parameters, as well as calculated measurements that contribute to the overall WSJF score.

WSJF fields	
Business Value	The feature's value to customers or the business. For example, how the feature will affect revenue.
Time Criticality	The urgency to deliver the feature in the release.
RR | OE	The feature's value in eliminating risks or creating new opportunities.
Calculated measurements	
Cost of Delay	The difference between a feature being available now or later. The Cost of Delay is the weighted sum of the above components: **Cost of Delay = (Business Value x BV weight) + (Time Criticality x TC weight) + (RR | OE x RR | OE weight)**
Job Size	The estimated length of time needed to implement the feature.
WSJF Score	The ratio of the Cost of Delay and Job Size. The higher the WSJF score, the higher the feature's priority. **WSJF Score = Cost of Delay / Job Size**

Setup: Configure WSJF settings

1. In Configuration, select **Workspace** > **Settings**.
2. Under **Weighted Smallest Job First (WSJF)**, select **Enable WSJF**. This makes the WSJF columns available in grids.
3. Set the weights (0 / 0.25 / 0.5 / 0.75 / 1) you want to assign to the Cost of Delay components. The weights influence the calculation of the features' Cost of Delay.

Setup: Add columns to the Features grid

1. Select the **Product Backlog** > **Features** tab.

2. Add the **Initial Estimate**, **Feature Type**, and **WSJF** columns to the grid.

Step 1 – Define a feature

Manage features on the **Product Backlog** > **Features** tab.

1. Click **Add Item**. Make sure **Feature** is selected in the **Type** box.
2. Type a **Name**, and assign the feature to a **Theme** and **Application**.

> **Note:**
>
> - A feature automatically inherits the applications that its backlog items are associated with. You cannot remove these applications from the feature .
>
> Applications inherited from its backlog items are listed in the **Applications** column in the Features grid.
> - Backlog items are not automatically associated with a feature's applications.

3. In the grid, select the **Feature Type**: **Architectural** or **Business**
4. Assign the feature an **Initial Estimate**. The initial estimate is measured in 'shirt sizes' (XL, L, M, S, XS). This provides you with a recorded feature size early on in the release, before you need or are able to provide more details.

Step 2 – Set WSJF values

Use WSJF values to help you prioritize features. Based on your policy, you can prioritize according to Business Value alone, the Cost of Delay (CoD) sum, or the overall WSJF score.

We recommend focusing on one WSJF component at a time, and set values for that component across all the planned features. That way you will be able to keep to a consistent scale for the component in question.

> **Note:** For the CoD components, select values on the Fibonacci scale, from 1 to 21.

Step 3 – Set story points

Estimate a feature in story points. This should be a rough estimate taking into account the feature's overall size. The feature's story point estimation is independent of the aggregated story points of the feature's backlog items. For example, you may initially estimate a feature at 40 story points. After the feature is broken down into user stories and story points are estimated for each user story, they may amount to 50 aggregated story points. Agile Manager stores both values separately.

Estimating story points at the feature level is useful for planning features to a release, and tracking the remaining release capacity.

Step 4 – Plan to release

Drag features to a release bucket, or select features and click **Planning**  . For more details, see "Plan a release" on page 92.

If you have many releases, click the link at the top of the planning pane (for example **46/52 Releases**). Enter text in the search bar to filter the releases listed, and select the releases you want to display in the planning panel.

> **Note:** Items that are filtered out can still be selected in the background. The filter only affects the items displayed, and does not affect any selections.

Step 5 – Manage feature life cycle

As work on a feature progresses, manage its life cycle by changing its status. Following are suggested definitions for each status:

Status	Definitions
New	The feature's initial state.
Defined	• An initial estimate is set. • A priority has been set (Business Value, Cost of Delay, or WSJF score). • Story points have been estimated.
Scoped	The feature is planned for a release, but work on the feature has not yet started.
In Progress	The feature is broken into user stories and assigned to a release.
Done	All the feature's backlog items are completed.
Accepted	The product owner or manager is satisfied with the feature's implementation.
Descoped	The feature was removed from a release plan.

Step 6 – Analyze feature progress

- On the **Features** page, use the progress column to track the progress of each feature.
- On the **Dashboard**, use widgets in the Feature category to analyze feature planning and progress.

Step 7 – Split feature

If you have not completed a feature and the release comes to an end, split the feature to move open backlog items to another release. Completed backlog items are left in the original release.

> **Note:** If you do not select a release for the new feature, all open user stories or defects (depending on your selection) will be moved to the new feature, but will have no release assigned.
>
> To ensure that you can view these backlog items in a release, define a release for the new feature.

Right-click a feature, and select **Split Feature**.

Build the product backlog

Create a product backlog of user stories and defects. The product backlog serves as a basis for planning release backlogs.

Define themes and features

Define your product themes and features. For details, see "Define themes" on page 85 and "Define and prioritize features" on page 87.

Create and manage backlog items

1. Open the **Product Backlog** > **Backlog** page.
2. Make sure the **Backlog Items View** is selected.
3. Perform the following actions:

Import backlog items

You can define user stories and defects in a file and import them into Agile Manager's product backlog. This is useful if you are migrating your backlog from another tool. For more details, see "Import backlog items" on page 25.

+ Add user stories

Add user stories to the product backlog.

> **Note:** The following special characters are not supported in user story names: **\ ^ ***

a. Click **Add Item**, and select **User Story** in the **Type** dropdown list.

b. If you work with templates, select a template to use. The user story will be created with built-in details, such as pre-defined tasks. Templates are managed by workspace administrators.

c. Estimate story points for each story. This helps you plan release and sprint backlogs according to their capacity. For more details, see "Balancing release and sprint workloads" on page 94.

d. Define details for the user story, such as feature or application, team, or priority. This enables you to group and filter backlog items, and track the progress of the product at various levels.

   To show a filtered list of features, first select an application. The list of available features filtered to display only those features assigned to the selected application.

Rank the backlog

A key factor in planning release and sprint backlogs effectively is arranging the backlog by rank. After items in the backlog are ranked, you can plan the highest ranking items to the next release or sprint backlog.

For details, see "Rank the backlog" on page 33.

Associate a user story with an application

Specify which of your applications a user story relates to.

Select a backlog item, and click **Set Application**.

The backlog item's feature and theme are automatically associated with the same application.

Break a user story

Breaking a user story is useful for dividing a large user story into two or more smaller parts. You can postpone some of the new stories to later sprints or releases, and even assign them to other features.

For details, see "Create a group story" on page 35.

Watch backlog items

Select user stories or defects that you want to watch, and click **More Actions** > **Add to My Watch List**.

- You will receive mail notifications upon status changes, or if the backlog item exceeds a storyboard time limit.
- View a summary of your watched items in the **My Watched User Stories** and **My Watched Defects** widgets.

View dependencies

If a backlog item has dependencies, an icon is displayed in the **Linked Items** column.

- Filter by the Linked Items column to display items with dependencies.
- Hover over the icon. A summary of the dependencies is displayed in the Linked Items section. Click **View All** to open a diagram of the item's dependencies.

Track the progress of backlog items

After development begins, use progress bars to track the progress of user stories and defects.

Make sure the **Progress** column is displayed in the grid.

- The amount of work done on a backlog item is measured by the time spent on the backlog item's tasks.
- The progress bar represents the hours invested in the backlog item's tasks compared with the remaining hours.

Next steps

Before the beginning of a release, use backlog items from the product backlog to plan a release. For details, see "Plan a release" below.

For a complete list of actions you can perform on backlog items, see "Backlog item actions" on page 50.

Plan a release

Push items from the product backlog to a release backlog. You can either plan features to a release, and drag their backlog items along with them, or plan individual backlog items.

This topic includes:

Release planning prerequisite

Set up and configure a new release. For details, see "Configure releases" on page 177.

Plan features to a release

When you are ready to work on a feature, assign it to a release.

1. Open the **Product Backlog** > **Features** page.
2. Either drag features to a release bucket, or select the features and click **Planning**.

   If you have many releases, click the link at the top of the planning pane (for example **46/52 Releases**). Enter text in the search bar to filter the releases listed, and select the releases you want to display in the planning pane.

   > **Note:** Items that are filtered out can still be selected in the background. The filter only affects the items displayed, and does not affect any selections.

For details on prioritizing features, see "Define and prioritize features" on page 87.

Plan backlog items to a release

When you assign a feature to a release, the Feature Planning dialog box opens, allowing you to specify which of the feature's backlog items will be assigned to the release along with the feature.

> **Note:** Only backlog items that are currently open will be assigned to the release. Completed backlog items, or backlog items that are added later will not be assigned.

Alternatively, you can assign individual backlog items to a release:

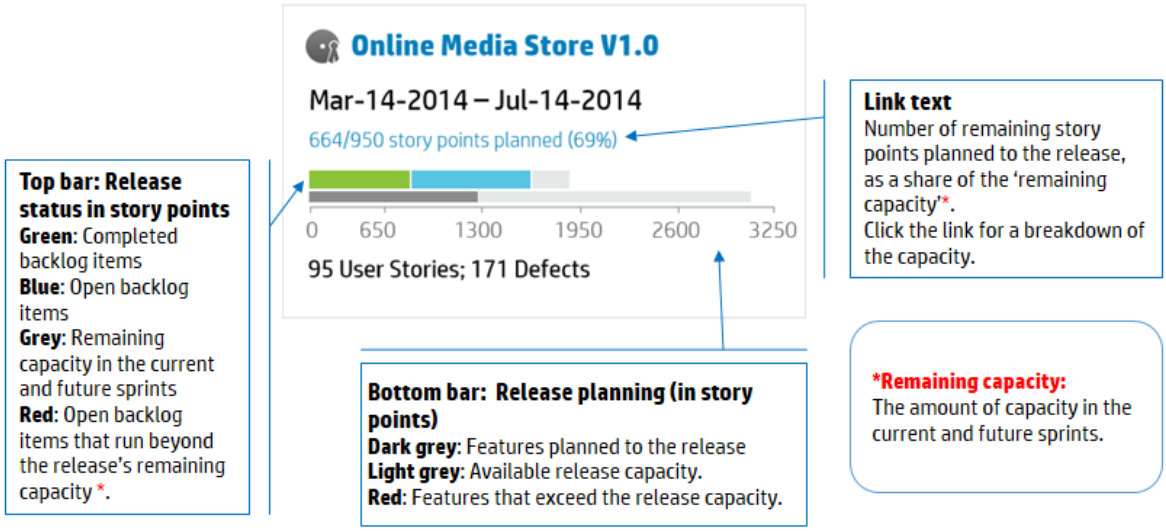
1. Open the **Product Backlog** > **Backlog** page.
2. Select the **Backlog Items View**.
3. Sort the grid by rank or priority.
4. Filter the grid according to your planning criteria. For example, define a filter to display user stories of a particular feature.
5. Select items in the grid, and drag them to a release bucket in the planning pane.

   If you have many releases, click the link at the top of the planning pane (for example **46/52 Releases**). Enter text in the search bar to filter the releases listed, and select the releases you want to display in the planning pane.
6. As you plan a release, pay attention to the release capacity. The story points of the items you plan should roughly match the release capacity in story points. The release buckets indicate the amount of story points assigned to the release and the remaining capacity. For more details, see "Balancing release and sprint workloads" on the next page.
7. To navigate to the release backlog you created, click the link in the release bucket.

> **Tip:** You can also plan backlog items directly to a sprint and team. Select items in the grid, and click **Planning**.

Understanding the release bucket

After you assign features or backlog items to a release, the release bucket is refreshed to reflect the new contents.



Balancing release and sprint workloads

Effective planning of release and sprint backlogs involves assigning the right amount of work items for the release capacity.

To facilitate this, each team is assigned a sprint velocity, which determines the overall release capacity. At the other end of the equation, the size of each user story and defect is estimated in story points. After completing these estimations, you can create an optimal release and sprint work plan.

This topic includes:

Set sprint velocities

The expected sprint velocity is the estimated number of story points a team can deliver in a sprint. Each team has a different velocity. The team velocities contribute to the following Agile Manager

measurements:

- **Sprint velocity.** The velocities of all the teams participating in a sprint determines the sprint velocity.
- **Release capacity.** The velocities of all the sprints determines the overall release capacity.

The sprint and release capacities serve as guidelines for the amount of backlog items you can plan to each release, sprint, and team.

After a team has participated in several sprints, use Agile Manager widgets to analyze the amount of story points the team managed to deliver. Based on this historical data, you can reevaluate the team's expected velocity for future sprints.

> **Note:** Team velocities are set by administrators on the team details page in the configuration area.

Set story points

Each backlog item – user story or defect – differs in size. In Agile methodology, you estimate the size of backlog items in **story points**. The amount of story points you assign to a work item denotes its size relative to other work items.

Optionally, an organization can decide to use story points as absolute values. For example, you could define that a user story with one story point requires one day of work to implement.

> **Note:** Agile Manager notifies you if a backlog includes items that do not have a story point estimation.

Plan release and sprint backlogs

Assign work items to the release or sprint backlog to fill the available capacity. Agile Manager graphically displays the amount of free space in a backlog, and warns if a backlog exceeds the capacity.

If a backlog has exceeded its capacity, use the following methods to balance the workload:

- Unassign user stories or defects from the release or sprint.
- Break large user stories into parts, and leave only a part in the current backlog.
- If feasible, increase the teams' velocities.

How to break a user story

Breaking a user story is useful for dividing a large user story into two or more smaller parts. You can postpone some of the new stories to later sprints or releases, and assign them to other features.

1. Select a user story, and click **Break Story**.
2. In the Break Story dialog box, define parts that constitute the original user story, and assign story points for each new story.
3. Click **Finish**. The new stories are added to the backlog.

Results:

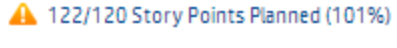
- The original user story becomes a group story. Use the **Group Story View** on the **Product Backlog** > **Backlog** page to view the group story and drill down to its children.
- The new stories inherit the original story's tasks, acceptance tests, and entity links.
- Story points assigned to the original user story are removed. The group story's story points are the sum of the story points assigned to the new user stories.

> **Note:** You can continue to break the new user stories into lower-level parts. These are all maintained under the same hierarchy as the original user story.

How to increase the sprint and release capacity

If a sprint or release is over-planned, you may want to increase its capacity. The sprint and release capacities are derived from the estimated sprint velocities per team, as defined in the team settings pages.

To adjust a team's sprint velocity:

1. On the **Product Backlog** or **Release Backlog page**, in a release or team bucket on the right, click a capacity information link 

.
2. Examine the actual velocity in previous sprints and the average velocity.

   In the release capacity or team velocity dialog box, use the bar chart to examine the expected and actual velocity in previous sprints, and compare them to the average velocity.

   Hover over the bars to view exact numbers.
3. If the average velocity is greater than expected, consider increasing teams' estimated velocities.

   If you're on the **Product Backlog** page, click the link on the bottom to the team settings page, and modify the velocity for the specific sprint there.

   If you're on the **Release Backlog** page, you can modify the team velocity directly in the dialog box, by double-clicking the number in the **Expected Velocity** column. The bar chart adjusts accordingly.

Break a user story

Breaking a user story is useful for dividing a large user story into two or more smaller parts. You can postpone some of the new stories to later sprints or releases, and assign them to other features.

1. Select a user story, and click **Break Story**.
2. In the Break Story dialog box, define parts that constitute the original user story, and assign story points for each new story.
3. Click **Finish**. The new stories are added to the backlog.

Results:

- The original user story becomes a group story. Use the **Group Story View** on the **Product Backlog** > **Backlog** page to view the group story and drill down to its children.
- The new stories inherit the original story's tasks, acceptance tests, and entity links.
- Story points assigned to the original user story are removed. The group story's story points are the sum of the story points assigned to the new user stories.

> **Note:** You can continue to break the new user stories into lower-level parts. These are all maintained under the same hierarchy as the original user story.

How to increase the sprint and release capacity

If a sprint or release is over-planned, you may want to increase its capacity. The sprint and release capacities are derived from the estimated sprint velocities per team, as defined in the team settings pages.

To adjust a team's sprint velocity:

1. On the **Product Backlog** or **Release Backlog page**, in a release or team bucket on the right, click a capacity information link 

.
2. Examine the actual velocity in previous sprints and the average velocity.

   In the release capacity or team velocity dialog box, use the bar chart to examine the expected and actual velocity in previous sprints, and compare them to the average velocity.

   Hover over the bars to view exact numbers.
3. If the average velocity is greater than expected, consider increasing teams' estimated velocities.

   If you're on the **Product Backlog** page, click the link on the bottom to the team settings page, and modify the velocity for the specific sprint there.

   If you're on the **Release Backlog** page, you can modify the team velocity directly in the dialog box, by double-clicking the number in the **Expected Velocity** column. The bar chart adjusts accordingly.

Archive backlog items

Administrators can archive themes and features to remove old items from the product backlog. This can increase your Agile Manager performance, as well as decrease the length of your lists of themes and features.

When you archive a theme or feature, all child items are also archived, including any items not marked as **Done**. Archived child items can include features (when archiving a theme), user stories, defects, or group stories.

Archived items hidden in	Archived items represented or displayed in
Backlog and release grids Sprint progress bars Group stories Task boards	Graphs and buckets for previous and current releases. These maintain accurate images of a specific release or sprint, even after relevant information is archived. Read-only display of archived themes, features, or backlog items.

> **Note:** Users can view a read-only list of archived items, but cannot retrieve them from the archive. Administrators must retrieve an item before its details can be displayed or modified.

Administrators can retrieve items from the archive, including individual user stories and defects.

- Retrieving a parent item (theme or feature) also retrieves all of its child features, user stories, or defects.
- Retrieving a child item (feature, user story, or defect) also retrieves its parent feature and theme. Other child items of the same feature and theme remain archived.

Retrieved items are visible in grids, and are accessible to users as they were before being archived.

Synchronized archived items

If an item in the archive is synchronized with ALM, any changes made to the item in ALM continue to be synchronized with the archived user story or defect.

What do you want to do?

Archive a theme or feature

> **Note:** You must be an administrator to archive items.

1. On the **Product Backlog** > **Themes** or **Features** page, select the items you want to archive.
2. Select **More Actions** > **Archive**.

View or retrieve archived items

> **Note:** You must be an administrator to retrieve items from the archive.

1. Do one of the following:

View archived items of a specific type	Open the **Product Backlog** > **Theme**, **Feature**, or **Backlog** page, depending on the type of items you want to view.
View archived child items for a specific theme or feature	Open the **Details** page for a specific theme or feature. For example, on the Features grid, click a feature ID to view its Details page. Click **Features** or **Backlog Items**, depending on whether you are looking at a theme or feature. All child items are displayed in a grid.

2. Select **More Actions** > **Display Archive...**.
3. Administrators only: To retrieve items, select the items you want to return to the backlog, and click **Retrieve from Archive**.

Product backlog reference

Following are descriptions of the entities that comprise the product backlog, and the relationships between them. The examples in this topic describe the development of a fictitious online store.

All product backlog entities are viewed within the context of a specific workspace. Enterprise organizations can use multiple workspaces for scaled agile projects. In such cases, users are able to view items in the workspaces to which they are assigned.

Users with access to multiple workspaces can toggle between them using the dropdown menu in the header. This dropdown menu is not displayed if there is only a single, default workspace defined in the site.

Applications

With Agile Manager you can manage multiple applications within the same workspace. Applications are the different components developed by your teams. Applications can be independent of each other, or

composite – designed to be integrated into a single product.

The user stories for all the applications are managed in a combined product backlog, providing you with visibility into the entire workspace's activity, and enabling you to define dependencies between user stories in different applications.

> **Application examples**
>
> The online store workspace may include the following applications:
>
> - Web Site
> - Mobile App
> - Marketplace

Themes

Themes are the top-level objectives your product must meet, or the product's high-level functional areas.

A theme is broken down into features. A theme may span several applications.

> **Theme examples**
>
> The online store workspace may list the following themes:
>
> - Music Store
> - Billing Module
> - Security Compliance

Features

Features are the functional areas of a product. Several features can be grouped together under a theme. A feature is broken down into user stories. You can associate features with multiple applications.

Agile Manager supports two types of features: **Business** and **Architectural**. Use a Business feature for customer facing changes, or an Architectural feature for changes required in your system to support a Business feature.

> **Feature examples**

The **Music Store** theme could include the following Business features:

- Album Database
- Song Playback
- Music Purchase

To support the Music Purchase feature, you might also develop a Secure Browsing feature.

Backlog items

The user stories and defects that are listed in the product backlog are known as backlog items. From the product backlog, you plan backlog items to release and sprint backlogs.

User stories

User stories describe the actions the user should be able to carry out by interacting with the product. Each user story should be associated with a feature, and may be associated with an application.

User story examples

The **Album Database** feature could include the following features:

- As a user, I can recommend a song to my friends
- As a user, I can rate an album I downloaded
- As a site administrator, I can delete an album from the album database

Grouped user stories

Grouped user stories are a set of related user stories, and are created by breaking an original user story into several children.

You can assign each child user story to different features, teams, or sprints, and even different releases. Child stories always inherit the parent story's tasks, acceptance tests, and entity links.

Grouped user stories are helpful when a large user story must be divided into multiple smaller parts, to be completed at different times and/or by different teams. You can track the group progress using the **Group Stories** view from the **Product Backlog** > **Backlog** page.

Grouped user story example

The online store workspace could include a group story for downloading MP3 files.

> Child user stories may be used to plan updates for various features in the Music Store and Billing themes. These child user stories could be developed over various sprints and releases, while ensuring that they have the same acceptance tests and can be tracked as a group.

Defects

The faults or bugs detected in the product. Defects are included in the product backlog together with user stories. Defects can be linked to user stories, and can be associated with features and applications.

Tasks

User stories and defects are broken down in to tasks.

When you create a new task, estimate the number of hours you think the task will require. As you complete the task, report your progress by updating the number of remaining hours in the Task Board. After you mark a task as completed, the task's remaining hours are automatically transferred to the invested hours.

- The estimated hours for all tasks in a backlog item are aggregated into the number of Planned hours for that item.
- The invested and remaining hours for all tasks are aggregated into the number of Actual hours for that item.

Compare the Planned and Actual hours for backlog items in the Details page, the Release and Sprint Backlog grids, or custom graphs.

> **Tip:** You can edit tasks to add additional, unplanned hours, or to update an estimate.
>
> Agile Manager does not retain the history of estimated hours, and aggregates only the current estimated value.

Product backlog Q&A

Q&A – Themes

How do I assign backlog items to a theme?

You cannot assign user stories or defects directly to a theme. You assign features to a theme. Backlog items assigned to a feature are indirectly associated with the feature's theme.

How can I search for specific items in the backlog?

You can search for items in the backlog by applying a filter to the grid. Create a filter using the grid column filter boxes, or the filter sidebar.

These are the expressions you can use on the different field types:

- **Text fields.** Enter part of a string. Use the ***** wildcard in the middle of the string to expand the search.
- **Number fields.** Enter a value to filter by. Use the following operators to expand the search:
  - Use one of the following operators to define an open-ended range: **>**, **<**, **>=**, **<=**, **<>**, and **""** to display all items with no value
  - Use a hyphen between low and high values to define a closed range.
- **List fields.** Select pre-defined values to filter by.

Where can I see which features and themes are planned for a release?

1. Select the **Release Management** > **Release Backlog** tab.
2. Make sure the correct release is selected.
3. Create a grouping of themes and features.

The user stories planned for the release are listed under their features and themes.

Am I required to define themes and features for my product?

You do not have to define your product themes and features in Agile Manager. Using themes and features enables you to track backlog items on a larger scale, and to gain visibility of the overall progress of the product.

How can I display a hierarchy of themes, features and backlog items?

The Features page displays a flat list of all product features, and the Backlog page displays a flat list all product backlog items.

To display features and backlog items in a hierarchical view:

- **Features page:** Define a grouping of themes. The features are grouped under the themes they belong to.
- **Backlog page:** Define a grouping of themes, and a sub-grouping of features. The backlog items are grouped under their themes and features.

Q&A – Features

How can I display a hierarchy of themes, features and backlog items?

The Features page displays a flat list of all product features, and the Backlog page displays a flat list all product backlog items.

To display features and backlog items in a hierarchical view:

- **Features page:** Define a grouping of themes. The features are grouped under the themes they belong to.
- **Backlog page:** Define a grouping of themes, and a sub-grouping of features. The backlog items are grouped under their themes and features.

How can I search for specific items in the backlog?

You can search for items in the backlog by applying a filter to the grid. Create a filter using the grid column filter boxes, or the filter sidebar.

These are the expressions you can use on the different field types:

- **Text fields.** Enter part of a string. Use the ***** wildcard in the middle of the string to expand the search.
- **Number fields.** Enter a value to filter by. Use the following operators to expand the search:
  - Use one of the following operators to define an open-ended range: **>**, **<**, **>=**, **<=**, **<>**, and **""** to display all items with no value
  - Use a hyphen between low and high values to define a closed range.
- **List fields.** Select pre-defined values to filter by.

How do I assign user stories and defects to features?

When you create a backlog item, you are prompted to assign it to a feature. If you did not do so while creating the backlog item, you can assign it to a feature on the Backlog page.

Where can I see which features and themes are planned for a release?

1. Select the **Release Management** > **Release Backlog** tab.
2. Make sure the correct release is selected.
3. Create a grouping of themes and features.

The user stories planned for the release are listed under their features and themes.

Am I required to define themes and features for my product?

You do not have to define your product themes and features in Agile Manager. Using themes and features enables you to track backlog items on a larger scale, and to gain visibility of the overall progress of the product.

Q&A – Backlog grid

I cannot find items in the product backlog.

Try the following:

- Check which favorite view is selected. It may have filtered out some of the backlog items you are looking for.
- Check the data filters. Click **Clear Filters** to remove all filters, or expand the filter pane to change the filter conditions.
- Check the application selector at the top of the screen. You must have permissions to access the application assigned to the backlog item, or the **(Not Set)** Application field value to access items with no defined application.
- The grid may be displaying only items without estimated story points. Check the status bar for a **Show All** link.

How can I display a hierarchy of themes, features and backlog items?

The Features page displays a flat list of all product features, and the Backlog page displays a flat list all product backlog items.

To display features and backlog items in a hierarchical view:

- **Features page:** Define a grouping of themes. The features are grouped under the themes they belong to.
- **Backlog page:** Define a grouping of themes, and a sub-grouping of features. The backlog items are grouped under their themes and features.

How can I find a backlog item's feature or theme?

Hover over the item's ID link or type icon (for example, in a grid). The tooltip that appears contains links to the item's feature and theme, if there are any defined.

Why is it important to estimate items in story points?

Story points indicate the relative size of backlog items. Release capacities and sprint velocities are measured in story points. Assigning story points to backlog items helps you plan release and sprint backlogs accordingly.

You may see a message in the status bar or in a planning bucket informing you that items are not estimated in story points.

In a planning bucket: Click the release name to navigate to the release backlog. Check the backlog items that have blank **Story Points**. Estimate their story points.

In the status bar: Click the **% not estimated** link in the status bar. The grid displays only those items that do not have story points. Estimate their story points.

> **Note:** The message in the status bar refers only to the number of non-estimated items in the current filter. You may receive a similar message after you change the filter settings.

What does the information in the status bar represent?

All the information in the status bar refers to the backlog items that are included in the current filter. A wheel  on the left of the status bar indicates that a filter is applied. Hover over the wheel to view the filter definition.

To view overall data for the entire backlog, clear all filters.

How can I search for specific items in the backlog?

You can search for items in the backlog by applying a filter to the grid. Create a filter using the grid column filter boxes, or the filter sidebar.

These are the expressions you can use on the different field types:

- **Text fields.** Enter part of a string. Use the ***** wildcard in the middle of the string to expand the search.
- **Number fields.** Enter a value to filter by. Use the following operators to expand the search:
  - Use one of the following operators to define an open-ended range: **>**, **<**, **>=**, **<=**, **<>**, and **""** to display all items with no value
  - Use a hyphen between low and high values to define a closed range.
- **List fields.** Select pre-defined values to filter by.

How can I reduce the amount of items displayed in the grid?

If you have a large number of user stories and defects in the grid, you can create a filter that displays a subset of the items. In addition, you can group items by values of a selected field. You can then expand and collapse groups as needed.

Who can modify a Public favorite?

Any user can modify a Public favorite. You do not need administrator privileges to do so.

Q&A – Release planning

There are warnings in the release planning pane. How do I address them?

You have planned more story points for a release than the release's capacity. The release capacity is based on the aggregate team velocities in all the release sprints.

Use the following methods to match the work load to the release capacity:

- If feasible, increase the teams' expected velocities. For details, see "How to increase the sprint and release capacity" on page 97.
- Unassign user stories or defects from the release.
- Break large user stories into parts, and leave only a part in the current backlog. For details, see "Break a user story" on page 36 above.

Can I plan backlog items directly to a sprint or team?

The suggested work method in Agile Manager is to first plan a release backlog. From the release backlog, you plan individual team sprints.

If you know in advance to which sprint or team particular backlog items are designated, you can plan the items directly.

1. Select items in the grid.
2. Click **Planning**.
3. In the Planning dialog box, select a **Release**. Then select a **Sprint**, **Team**, or both.

I cannot find a release in the planning pane.

Try one of the following:

- Use the vertical arrows to scroll up and down the release buckets.
- The release may be filtered from the list. Click the release filter link at the top of the pane to select additional releases.

> For example, if there are a total of two releases in the workspace, but only one is currently displayed, this filter reads: **1/2 Releases**

Release management

Build and manage a release backlog, track the release progress, and analyze its quality in the **Release Management** area of Agile Manager.

If you use a Kanban storyboard to control your development process throughout the release, see "Kanban storyboard" on page 131 for details.

This section includes the following topics:

Manage the release backlog

Before a release begins, create a backlog of the user stories that you plan to handle in the release.

Prerequisite: Set up and configure a new release. For details, see "Configure releases" on page 177.

1. **Plan a release from the product backlog.** On the **Product Backlog** > **Backlog** page, drag backlog items into the release bucket, based on your planning strategy. For more details, see "Plan a release" on page 92.
2. Open the **Release Management** > **Release Backlog** page.
3. Make sure the correct release is selected.
4. Perform the following actions:

Import backlog items

You can define user stories and defects in a file and import them directly to a release in Agile Manager. This is useful if you are migrating your backlog from another tool. For more details, see "Import backlog items" on page 25.

Add user stories to the release backlog

In addition to the backlog items you assign to the release from the product backlog, you can add user stories directly to the release backlog:

a. Click **Add Item**, and select **User Story** in the **Type** dropdown list.

b. If you work with templates, select a template to use. The user story will be created with built-in details, such as pre-defined tasks. Templates are managed by workspace administrators.

c. Estimate story points for each story. This helps you plan release and sprint backlogs according to their capacity. For more details, see "Balancing release and sprint workloads" on page 94.

d. Define details for the user story, such as feature or application, team, or priority. This enables you to group and filter backlog items, and track the progress of the product at various levels.

Backlog items you add to a release backlog are included in the product backlog.

Report defects

a. Click **Add Item**, and select **Defect** in the **Type** dropdown list.

b. If you work with templates, select a template to use. The defect will be created with built-in details, such as pre-defined tasks.

> **Tips:**
>
> - You can create a defect and link it directly to a user story or to another defect. Select an item, and click **More Actions** > **Create Linked Defect**.
> - Use the **Defects Management** page to take advantage of defect tracking and management tools.

Rank the backlog

A key factor in planning release and sprint backlogs effectively is arranging the backlog by rank. After items in the backlog are ranked, you can plan the highest ranking items to the next release or sprint backlog.

For details, see "Rank the backlog" on page 33.

View dependencies

If a backlog item has dependencies, an icon is displayed in the **Linked Items** column.

- Filter by the Linked Items column to display items with dependencies.
- Hover over the icon. A summary of the dependencies is displayed in the Linked Items section. Click **View All** to open a diagram of the item's dependencies.

Break a user story

Breaking a user story is useful if you need to divide a large user story into two or more smaller parts. You can postpone some of the new stories to later sprints or releases, and even assign them to other features.

For details, see "Create a group story" on page 35.

Watch backlog items

Select user stories or defects that you want to watch, and click **More Actions** > **Add to My Watch List**.

- You will receive mail notifications upon status changes, or if the backlog item exceeds a storyboard time limit.
- View a summary of your watched items in the **My Watched User Stories** and **My Watched Defects** widgets.

Track the progress of backlog items

After development begins, use progress bars to track the progress of user stories and defects.

Make sure the **Progress** column is displayed in the grid.

- The amount of work done on a backlog item is measured by the time spent on the backlog item's tasks.
- The progress bar represents the hours invested in the backlog item's tasks compared with the remaining hours.

Plan a sprint backlog

Before each sprint, drag backlog items into sprint buckets in the planning pane. For details, see "Plan team sprints" on page 112.

For a complete list of actions you can perform on backlog items, see "Backlog item actions" on page 50.

Use the planning board

Select the **Release Management** > **Planning Board** tab.

The planning board enables you to visualize and manage the release backlog, as well as plan sprint backlogs.

It's provides you with a bird's eye view of your release backlog plan and progress, and assists your release strategy planning. The planning board displays all the release backlog items in a configurable grid. Backlog items are represented on the grid by tiles.

You can design the Planning Board according to your planning strategy, by selecting the properties that define the grid's rows and columns. For example, the grid **columns** could list sprints, and the **rows** – priorities. By examining how the tiles are distributed in the grid, you can evaluate where your high priority user stories are planned in the release.

In addition, you can select a property for a third dimension – **color**. Using the same example, if colors represent statuses, you can understand what progress has been made on the high priority stories in each sprint.

You can shift tiles on the planning board to dynamically change backlog item properties. For example, you can move a low priority user story to a later sprint, or you can even increase a user story's priority by moving its tile to a higher row.

Select from a number of preset combinations of columns, rows, and colors, or design a custom view.

> **Note:** By default, the Planning Board displays only 200 items at time. To display more items, click **Load More** at the top of the page.

What do you want to do?

Load a predefined view

Click the **Favorites** arrow, and select a view from the Private or Public folder.

The following preset favorites are available in the **Public** folder:

Preset Favorite	Suggested Use
Release backlog life cycle by theme	Track the progress of your product themes in the current release.
Sprint backlog planning by theme	Plan release backlog items to sprints based on theme.
Release backlog life cycle by application	Track the progress of your applications in the current release.
Release timeline mode by application	Plan release backlog items to sprints based on application.
Sprint backlog planning by type	Balance the allocation of user stories and defects to sprints and teams.
Sprint backlog planning by priority	Plan release backlog items to sprints based on priority.
Sprint backlog overview by status	Track the status of backlog items along the sprint timeline.

Design a custom view

1. In the **Columns**, **Rows**, and **Color** boxes, select the properties that will define the grid's dimensions.

   Besides the default properties, you can select also customized single-selection list fields.
2. Click the **Favorites** arrow, and select **Add to Favorites**. Custom views that you save under the **Public** folder are available to all users.

Filter the Planning Board

By default, the Planning Board displays tiles for all the release backlog items. If you want to reduce the amount of tiles on the Planning Board, you can define a filter for user stories, defects, or both.

Expand the filter left sidebar, and select properties to filter by.

Review the progress of your release

Use the Planning Board to gain a picture of the progress of the release plan. You can achieve this by viewing the amount of backlog items in each status.

Click **Favorites**. From the Public folder, select the **Release life cycle by theme** view.

Each column in the view represents a status, and the rows represent teams. You can review the overall status of the release, and assess whether backlog items can be redistributed between teams.

In addition, the user story themes are represented by colors. This enables you to identify the progress made on particular themes.

Plan sprints with the Planning Board

Use one of the following options:

- Select a view that lists sprints and teams along the columns and rows of the planning board. Backlog items that are not planned to a sprint or a team are located in the **No value** boxes at the end of the sprints and teams lists. Drag tiles from the **No value** boxes to a sprint and team box.
- Right click a tile, and select **Planning**.

Plan team sprints

Before a sprint begins, fill your team's sprint backlog with items from the release backlog.

Plan sprint backlogs by dragging items from the release backlog on the **Release Backlog** page. Or use the **Planning Board** for a bird's eye view of the sprint plans for the entire release, and shift items between sprints.

Plan sprints on the Release Backlog page

1. Open the **Release Management** > **Release Backlog** page.
2. Make sure the correct release is selected.
3. Sort the grid by rank or priority.
4. Filter the grid according to your planning criteria. For example, define a filter to display user stories of a particular feature.
5. On the right planning pane, select the sprint you want to plan. The teams participating in the sprint will be listed.
6. Select items in the grid, and drag them to your team bucket in the planning pane.

If you have many teams, click the link at the top of the planning pane (for example **10/12 Teams**). Enter text in the search bar to filter the teams listed, and select the teams you want to display in the planning pane.

7. As you plan a sprint, pay attention to the team's expected sprint velocity. The story points of the items you plan should roughly match the team's estimated sprint velocity. The team buckets indicate the amount of story points assigned to the team and the remaining velocity. For more details, see "Balancing release and sprint workloads" on page 94.
8. To navigate to the team sprint backlog you created, click the team link in the planning pane.

Plan the release with the Planning Board

The Planning Board is a versatile tool that enables you to visualize the release backlog in three dimensions. With it, you can view the distribution of release backlog items in selected attributes, such as priority, application and theme, and plan sprints according to your chosen criteria.

For details of working with the Planning Board, see "Use the planning board" on page 110.

Analyze a release

Agile Manager provides you with various analysis tools to track the progress and quality of a release.

Release Backlog page widgets

Use the widgets at the top of the Release Backlog page to review user story and defect statuses. The graphs refer to the items in the current filter.

Release Manager dashboard

On the **Dashboard**, select the predefined **Release Manager Dashboard** favorite. It includes the following widgets:

Release Burn Down

Displays the *story points* remaining to complete the release backlog, compared to the available capacity, at different stages of the release.

The straight diagonal line shows the remaining capacity (in hours) until the end of the release.

The curved line shows the hours of tasks remaining in the release.

- If the curved line is higher than the straight line, the team is behind schedule.
- If the curved line is lower than the straight line, the team is ahead of schedule.

Group Velocity

Displays the number of *story points* completed by the group in each sprint, compared to the group's average sprint output.

Sprints with a velocity that was significantly different from the average may be worth further analysis.

Release Defects Cumulative Flow Diagram

Displays the number of *defects* assigned to the group, grouped by status, at different stages of the release. Each band shows the number of user stories in a status.

Following are the ideal trends you should see in these graphs and in each of their statuses:

Total	In an ideal release, the shape of the graph should be almost rectangular, although there may be an increase of new items at the beginning of the release. At the beginning of the release, the graph's height is made up primarily of new items. From that point on, the planned items move through the different statuses, until they are all done.
New	Most new content should be planned at the beginning of the release. Increases in the **New** band indicates user stories that were added during the release. The band should gradually get narrower as user stories move to **In Progress**.
In Progress	The number of **In Progress** user stories should be more-or-less constant. The team should be closing up user stories, and moving them to **In Testing**, before starting work on additional items.
In Testing	The progress of the **In Testing** band should mirror the **In Progress** band, with a slight time lag. To ensure a gradual flow of items from **In Progress** to **In Testing**, the team should attempt to complete several small user stories shortly after each sprint begins.
Done	The **Done** band should grow steadily throughout the release, until it dominates the graph by the end of the release.

Release Backlog Cumulative Flow Diagram

Displays the number of *user stories and defects* assigned to the group, grouped by status, at different

stages of the release.

For a retrospective of the release planning, open the widgets gallery, and select widgets from the **Retrospective** category.

ALI analysis

For a presentation of the release quality based on source code and build information, open the **Release Management** > **ALI Summary** page.

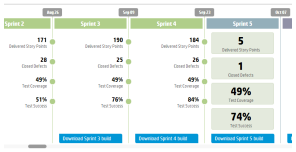
For more details, see "ALI: Analyze release quality" below and "ALI: Analyze application quality " on the next page.

ALI: Analyze release quality

Use ALI analytics to evaluate the quality of a release, based on build and source code information.

In the **Release Management** > **ALI Summary** page, verify that **Release Summary** is selected on the top left corner of the page.

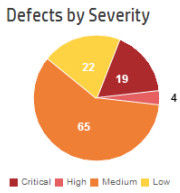
This page displays the following widgets:



Sprints Overview

Displays a summary of key statistics for each sprint along the release timeline.

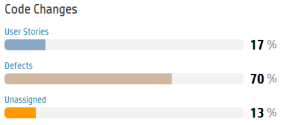
Use the horizontal scroll bar to view other parts of the release.



Defects by Severity

The number of open defects associated with the release, grouped by severity.

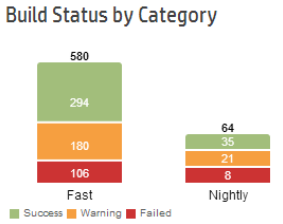
You can find more information about defects in the charts on the Agile Manager Dashboard. For more details, see "Dashboard and analysis tools" on page 65.



Code Changes

The percentage of code changes in the release associated with user stories, defects, or neither.

Click a category to open the Source Code Summary page to view the associated code changes. For more details, see HP ALI Q&A.



Build Status by Category

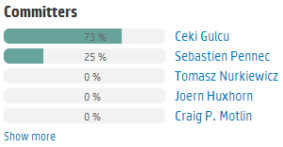
A breakdown of build status for each build type.

Click a configuration category to open the Build Summary page for that type. For more details, see Build Summary Page.

Development Alignment

Shows each user story that was worked on during the release and how many lines of code were changed for each story.

The user stories are displayed in descending order of priority (High, Medium, and Low). Within each priority, the stories are displayed in descending number of code changes. Click the **Code Changes** header to reverse this order.



Committers

A summary of the developers who changed and committed code in the release, sorted by level of contribution. The level of contribution is measured in Lines of Code (LOC).

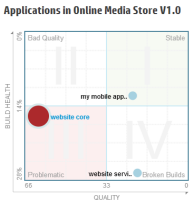
ALI: Analyze application quality

Use ALI analytics to evaluate the quality of applications in a release, based on build and source code information.

In the **Release Management** > **ALI Summary** page, verify that **Application Summary** is selected on the top left corner of the page.

All information on the page is specific to a time frame. Use the **Time period** drop down to select a time period.

This page displays the following widgets:



Quadrant chart

Plots your applications in two dimensions: **open defects** and **build results**. A third dimension – **code changes** – is represented by the size of the application bubbles.

Evaluate an application according to the quadrant in which it is located:

I - Stable: The application has an above average build success rate, and has few open defects.

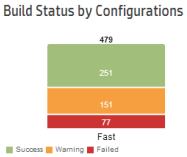
II - Bad Quality: The application has an above average build success rate, but the number of its open defects is above average.

III - Problematic: The application has a high build failure rate, and an above average number of open defects.

IV - Broken Builds: The application has few open defects, but a high build failure rate

Tip:

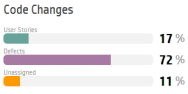
- Hover over an application's bubble to display its exact build failure rate and number of open defects.
- Click an application's bubble to change the application in context on the right side of the page.



Build Status by Configurations

Side-by-side charts displaying an application's build success and failure rates for each build configuration.

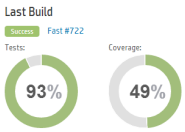
- Select an application from the drop down list.
- Click a build configuration chart to display builds belonging to the build configuration on the Builds page.



Code Changes

The percentage of the application's code changes associated with user stories, defects, or neither.

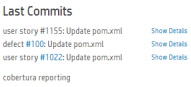
Select an application from the drop down list.



Last Build

Displays the build status, test and coverage information for an application's last build.

- Select an application from the drop down list.
- Click the build number to navigate to the build's details on the **Builds** page.



Last Commits

Displays comments for the last three commits associated with an application.

- Select an application from the drop down list.
- Click **Show Details** to navigate to the code change details on the **Source Code** page.

Applications grid

Compares key statistics on defects, last build results, and code changes for all applications.

- Select an application from the drop down list.
- Click a column header to sort the table by that column.

Sprint management

Plan team sprints, break work items into tasks, execute and analyze a sprint.

Prepare

Plan user stories and defects to a particular sprint or release.

For each item, define tasks and acceptance tests, and ensure that the total story points is equal to the team capacity.

Allocate items to team members to get started.

See "Prepare for a sprint" below.

Run

Run the sprint using the Task Board or the Storyboard, depending on the Agile methodology your team uses.

Complete tasks and acceptance tests, and handle any linked defects, as you move user stories to **Completed**.

Throughout your sprint, use graphs and widgets on the Dashboard to analyze your progress.

Based on the information you see, you may want to modify priorities, assignments, or other data.

See "Use the Task Board" on page 123 and "Kanban storyboard" on page 131.

Track and analyze

At the end of your sprint, take a look at the items your team completed.

List some practices your team did well, as well as some ways you can perform better next time.

Handle open items by moving them or splitting them to the next sprint.

See "Track and analyze the sprint" on page 126, "Split a user story" on page 128, and "Sprint retrospective and closure" on page 130.

Prepare for a sprint

At the sprint planning meeting, create a backlog of user stories and defects you plan to handle in the sprint, and define tasks for the backlog items. Then assign backlog items and allocate tasks to team members.

1. Open the **Release Management** > **Sprint Backlog** page.
2. Make sure the correct sprint and team are selected.

What do you want to do?

Plan a sprint backlog

Use the following pages to plan sprint backlogs:

- **Release Backlog:** Plan sprint backlogs by dragging items from the release backlog into sprint buckets. For details, see "Plan team sprints" on page 112.
- **Planning Board:** Use the Planning Board for a bird's eye view of the sprint plans for the entire release, and shift items between sprints. For details, see "Use the planning board" on page 110.

You can also add unplanned user stories directly to the sprint backlog. Click **Add Item**.

Rank the backlog

The backlog items are ranked in the sprint backlog according to their ranks in the release backlog. Sort the sprint backlog by rank, and adjust the ranks as necessary.

For details, see "Rank the backlog" on page 33.

Define tasks

Backlog items are usually written from a user perspective. In order to create a development work plan, you need to break the user stories and defects into detailed tasks, written from a developer perspective. Tasks are displayed in the pane below the backlog grid.

Drag tasks up or down to organize them, such as in the order they should be performed.

1. Right-click a backlog item, and select **Add Task**.
2. Fill in the following details for each task:

Task	A description of the task.
Assignee	The user assigned to the task. By default, the owner of the backlog item is selected. You can assign the task to any member of the team.
Estimation	Tasks are measured in hours, as opposed to backlog items that are estimated in story points. Specify the number of hours you estimate it will take to complete the task. Administrators define default task estimations on the **Workspace** > **Settings** configuration page.
Color Category	Optionally, assign the task to a color category. For example, use color categories to identify development, QA, and documentation tasks.

Define acceptance tests

Define the criteria that a user story must meet in order to be considered complete. Acceptance tests are displayed in the pane below the backlog grid.

According to your workspace settings, a user story may not to be set to **Done** until all its acceptance tests have passed.

Allocate work items to team members

Before you allocate tasks to team members, you should calculate each team member's capacity.

Then, assign backlog items and allocate tasks to team members by dragging them to user buckets in the planning pane on the right.

Task	Description
Compare assigned hours vs. capacity	The top of the planning pane indicates whether you are planned within your team capacity () or over-planned (), by displaying: the **total assigned hours for the team sprint** vs. the **total capacity of all team members**. Total assigned hours includes both invested and remaining hours. Each team member bucket indicates the amount of hours assigned vs. the total capacity defined for that team member. Click the capacity link for each team member  to modify that team member's capacity: **No. of Work Days** The number of days the team member is available to work on the team in the current sprint. This figure reflects the number of days in the sprint, deducting weekend days. If the team member has planned vacation or leave days during the sprint, reduce the number of working days accordingly. **Hours per Day** The number of hours per day the team member is available to work on the team. If the user is a member on several teams, adjust the hours according to the time the team member is available on the current team.
Assign backlog item	To set the owner of a backlog item, select the item and drag it to the user's bucket. All the backlog item's tasks are added to the owner's workload. Later, you can reallocate specific tasks to other users.

Task	Description
Assign tasks	When defining tasks, you can assign them to a user. Later, you can reallocate tasks by dragging them to user buckets in the planning pane. **Note:** If a backlog item was reassigned to another team, tasks may still be allocated to users in the original team.

Export a sprint backlog

Export a sprint backlog grid to an Excel or CSV file to use sprint data in other applications. Exported data includes any columns and filters displayed on the grid, and the **Description** and **Comments** fields.

Tip: If you are exporting to Excel, you can also include tasks and acceptance tests.

1. Define a filter for the items you want to export and arrange columns you want to include.
2. Click **More Actions** > **Export Backlog to Excel** or **Export Backlog to CSV**.
3. In the **Export Backlog to Excel** dialog box, select whether you want to include tasks and acceptance tests.

   Tasks and acceptance tests are exported to separate Excel tabs.

Next steps

Track sprint execution

Use the following Agile Manager tools to track the sprint execution:

- **Task Board.** Use the task board to track the progress on each of your tasks, and report the time invested in them. For details, see "Use the Task Board" on the next page.
- **Storyboard.** Use the storyboard to visualize and enforce your team processes on the development of backlog items. The storyboard can be used within the boundaries of a single sprint, or across multiple sprints. For details, see "Kanban storyboard" on page 131.

Analyze sprint progress and performance

- During the sprint, use varied widgets to analyze the sprint progress. For details, see "Track and analyze the sprint" on page 126.
- At the end of the sprint, perform a retrospective of the sprint. For details, see "Sprint retrospective and closure" on page 130.

Use the Task Board

Track the progress of your team's backlog items and tasks during a sprint. Items on the task board are listed in order of their rank.

1. Open the **Release Management** > **Task Board** page.
2. Make sure the correct sprint and team are selected.

> **Tip:** The task board continues to load as you scroll down the page. To reduce the time it takes to load, use a filter to limit the items displayed.

Load all items at once

If you want to load all items at once instead of as you scroll, select **Load All** above the grid. Clear the **Load All** check box to return the default behavior.

Your selection remains for the duration of your current Agile Manager session.

Filter the task board

The task board initially displays all the backlog items and tasks planned for the selected team and sprint. If you have many items, the Task Board continues to load as you scroll down the page.

To reduce the number of items displayed on the task board, Agile Manager provides you with the following filtering options:

Display backlog items and tasks assigned to you	In the **Assigned To** box, select your user name. The following backlog items are displayed: • Backlog items assigned to you, even if none of its tasks are allocated to you. • Backlog items with tasks assigned to you. Tasks assigned to other users are grayed out.
Display tasks assigned to users on other teams	If a backlog item was reassigned to you from another team, its tasks may still be assigned to members on the original team. To display such tasks, in the **Assigned To** box, select users listed on the bottom part of the list, below the members on your team.
Toggle user stories and defects	Use the **User Stories** and **Defects** check boxes to display only user stories or defects, or both.

Display specific backlog items	Enter text in the filter box, located on the top left of the page. Only backlog items that include the search text will be displayed.
Hide done backlog items	To clear done backlog items from the task board, in the **Backlog Items in Status** box, deselect **Done**.
Display backlog items without tasks	To display only backlog items that have no tasks, in the **Assigned To** box, select **Items Without Tasks**.

Export items to Excel

1. Define a filter for the items you want to export, and click **Export to Excel**.
2. In the **Export Backlog to Excel** dialog box, select the additional item(s) you want to export.

The displayed items are exported to an Excel workbook. Tasks and acceptance tests are listed on separate tabs.

Add a task to a backlog item

During the sprint, you may realize that additional tasks are required to complete the user story or defect.

Right-click an item in the **Backlog Item** column, and select **Add Task**. You can drag tasks around to organize them, such as in the order they should be performed.

Fill in the following details for each task:

Field	Description
Task	A detailed description of the task.
Assignee	The user assigned to the task. By default, the owner of the backlog item is selected. You can assign the task to any member of the team.
Estimation	Tasks are measured in hours, as opposed to backlog items that are estimated in story points. Specify the number of hours you estimate it will take to complete the task. **Note:** Administrators set default task estimates on the **Workspace** > **Settings** configuration page.
Color Category	Optionally, assign the task to a color category. For example, use color categories to identify development, QA, and documentation tasks.

Set planned task estimates and report actual task times

Before starting a task, estimate the number of hours of effort you think the task will require. As you progress, report time you spent working on tasks to provide a reliable picture of your sprint work effort.

The planned and actual time invested in tasks is also used in the team's sprint burn up and burn down graphs.

- Task times are measured in hours.
- Actual time refers to the sum of invested and remaining time.

Report time spent on a task	Click the **Remaining work** icon  , and click the **Decrease remaining** button  . As a result, the task's **Invested** time and the backlog item's **Actual** hours increase by the same amount.
Report time spent on a task, while leaving the amount of remaining work unchanged	Double-click the task, and increase the **Invested** value. As a result, the backlog item's **Actual** hours increase by the same amount, and the **Remaining** amount remains unchanged.
Modify a task's time estimation	Double-click the task and increase or decrease the **Estimated** value. **Note:** Modifying the task's time estimation replaces the original value. Graphs that analyze the Planned and Actual effort use the current **Estimated** time value.

Change a task status and close a task

To change the task status, drag the task to the **In Progress** or **Completed** column. Alternatively, right-click the task and select **Mark as Completed**.

When all the tasks defined for a backlog item are completed, you are prompted to change the backlog item's status.

To apply the same status to all backlog items when their tasks are completed, select the **Always use this status** check box.

Tip: To change the status applied to backlog items when all their tasks are completed, click **Ctrl+1**, and select or type **Reset 'Don't show this again' Selections**. The next time you complete all of a backlog item's tasks you will be prompted again to select a status.

Close a backlog item

A backlog item is usually closed automatically after all its tasks are completed. For more details, see "Change a task status and close a task" on the previous page above.

To manually set a backlog item to **Done**, right-click the backlog item and select **Set to Done**. By setting a backlog item to Done, all its tasks are automatically set to **Completed**, and the remaining hours are transferred to the invested hours.

> **Tip:** At the end of a sprint, all user stories should be closed. If you invested time in a user story, but not all its tasks and acceptance tests are completed, split the story. Remaining effort and non-passed acceptance tests are moved under a new user story, which you can assign to a future sprint.
>
> For further details, see "Split a user story" on page 128.

Analyze sprint progress and performance

- During the sprint, use varied widgets to analyze the sprint progress. For details, see "Track and analyze the sprint" below.
- At the end of the sprint, perform a retrospective of the sprint. For details, see "Sprint retrospective and closure" on page 130.

Track and analyze the sprint

Agile Manager provides you with a variety of analysis tools to use during your daily scrum meetings and at the sprint retrospective.

Sprint retrospective tools

- Use the widgets at the top of the **Release Management** > **Sprint Closure** page for a summary of the sprint.
- On the **Dashboard**, select widgets from the **Retrospective** category.

Daily sprint tracking

At the daily scrum meetings, use the **Scrum Master Dashboard** for an up-to-date snapshot of the sprint. It includes the following widgets:

Sprint Burn Down

Displays the work remaining to complete the team's sprint backlog, compared to the available team capacity, at different stages of the sprint.

The straight diagonal line shows the remaining team capacity (in hours) until the end of the sprint.

The curved line shows the hours of tasks remaining in the sprint.

- If the curved line is higher than the straight line, the team is behind schedule.
- If the curved line is lower than the straight line, the team is ahead of schedule.

Sprint Cumulative Flow Diagram

Displays the flow of the sprint's user stories through the different statuses, along the sprint time line. Each band shows the number of user stories in a status.

Following are the ideal trends you should see in the graph and in each of its statuses:

Total	In an ideal sprint, the shape of the graph should be almost rectangular. At the beginning of the sprint, the graph's height is made up primarily of new items. From that point on, the planned items move through the different statuses, until they are all done.
New	All new content should be planned by the beginning of the sprint. An increase in the **New** band means that user stories were added during the sprint. The band should gradually get narrower as user stories move to **In Progress**.
In Progress	The number of **In Progress** user stories should be more-or-less constant. The team should be closing up user stories, and moving them to **In Testing**, before starting work on additional items.
In Testing	The progress of the **In Testing** band should mirror the **In Progress** band, with a slight time lag. To ensure a gradual flow of items from **In Progress** to **In Testing**, the team should attempt to complete several small user stories shortly after the sprint begins.
Done	The **Done** band should grow steadily throughout the sprint, until it dominates the graph by the end of the sprint.

Team Velocity

Displays the number of story points completed by the team in the previous sprints, and in the current sprint to-date, compared to the team's average sprint output.

Team Workload

Displays the work remaining in the sprint for individual team members, compared to the member's remaining capacity.

Split a user story

At the end of a sprint, all user stories should be closed. If you invested time in a user story, but not all its tasks and acceptance tests are completed, split the story. Remaining effort and non-passed acceptance tests are moved under a new user story, which you can assign to a future sprint.

Available from: Sprint Backlog, Task Board, Sprint Closure (Open Items grid)

1. Right-click a user story, and select **Split Story**.
2. Specify the sprint that the new user story will be assigned to, and update details of tasks and acceptance tests as needed.

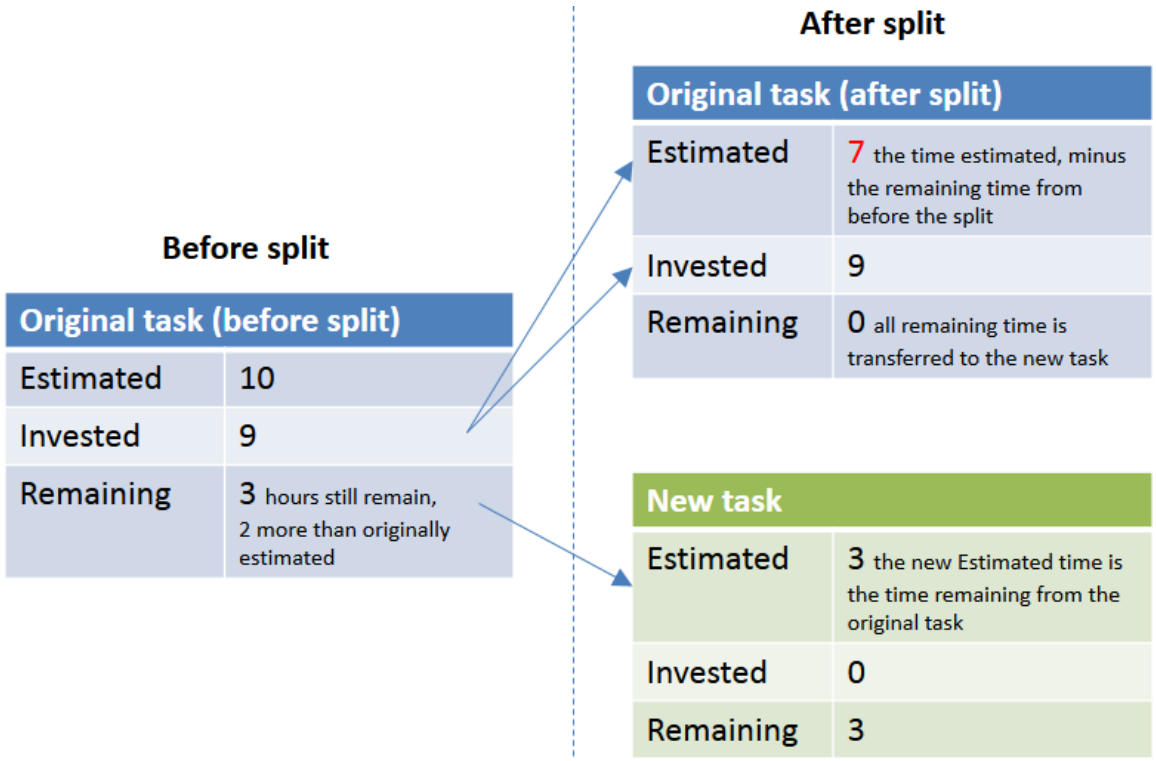
> **Note:**
>
> - You can only split a story within a release. You cannot split a story to a sprint in a different release.
> - Splitting a story is different than breaking a user story, which is done when you have a large story you want to divide into smaller pieces.
>
> After breaking, the group stories are visible only in the **Group Stories** view on the **Product Backlog** grid. For details, see "Create a group story" on page 35.

What happens when you split a user story?

	Original user story (after split)	New user story
Story points	Set to zero.	All the story points from the original story are transferred to the new story.
User story status	By default, set to **Done**. You can also keep the original story open after the split. Specify a different status in the Split Story dialog box.	**New**

Assigned to	Both user stories and tasks remain assigned to the users they were assigned to in the original story.	
Acceptance tests	**Passed** acceptance tests remain in the original story.	By default, all **Not Started** and **Failed** acceptance tests are moved to the new story. Clear any acceptance tests that you do not want to move over.
Tasks	• **Completed** tasks, and their invested time, remain in the original user story. • By default, **New** tasks are moved to the new user story. Clear any tasks that you do not want to move over. • By default, **In Progress** tasks are split between the original story and the new story. The invested time remains in the original task, and the remaining time is moved to the new task. You can choose not to split **In Progress** tasks. Clear the **Split tasks with invested effort** option. **In Progress** tasks will be carried over, with their invested time, to the new user story.	

Splitting time values when splitting tasks



Sprint retrospective and closure

When a sprint ends, use the **Sprint Closure** page to review a summary of the sprint, perform a sprint retrospective, and carry over open items to the next sprint.

Open the **Release Management** > **Sprint Closure** page, and make sure the correct release, sprint and team are selected.

What do you want to do?

Review sprint metrics

The dashboard at the top of the page includes the following reports:

Sprint Backlog Status	A summary of essential statistics regarding the team sprint.
Backlog Planning and Delivery	A graph comparing the amount of user stories included in the original sprint plan, and those added to the plan after the beginning of the sprint. Use the graph to evaluate how well you planned the sprint in advance.
Defect Cumulative Flow Diagram	Displays the flow of the team's defects through the different statuses.
Acceptance Tests	A summary of the passed and failed acceptance tests connected to the sprint's user stories.

For additional retrospective graphs on the sprint planning, open the **Dashboard**, and select widgets from the **Retrospective** category.

Write up a retrospective of the sprint

In the **Things that went well** and **Things to improve** boxes, list the team's positive and negative takeaways from the sprint.

If you created a bulleted or numbered list, you can automatically convert the items to action items. Exit the text box to save the text, and click **Convert to action item(s)**.

Create a list of action items

In the **Action Items** pane, create a list of the action items that resulted from the sprint retrospective. The action items are specific to the team, but not to the selected sprint.

Optionally, use the **Create Backlog Item** button to convert selected action items to backlog items for future sprints.

Resolve open backlog items

Ideally, at the end of a sprint all backlog items planned for the sprint should be closed.

1. Click the **Open Items** tab « located on the right of the page. A grid opens listing the leftover backlog items from the current sprint.

   You can edit the backlog items directly in the grid.

2. Use the following toolbar commands to close the open items:

Roll to Next Sprint	Move the selected backlog items to the next sprint.
Planning	Reassign the selected backlog item to another release, sprint, and team.
Split Story	At the end of a sprint, all user stories should be closed. If you invested time in a user story, but not all its tasks and acceptance tests are completed, split the story. Remaining effort and non-passed acceptance tests are moved under a new user story, which you can assign to a future sprint. For further details, see "Split a user story" on page 128.

Kanban storyboard

Manage your development life cycle on a fully configurable and interactive storyboard.

Use the storyboard to manage the flow of backlog items through their stages of development. Design the storyboard to reflect your team's work process, and shift backlog items along the storyboard to track their progress.

With a fully interactive and configurable storyboard, you can now work in full Kanban mode, or in Scrum mode enhanced by a storyboard workflow and Kanban metrics.

Storyboard design

The storyboard includes three predefined columns: **Planning** – **In Progress** – **Done**. These columns are fixed and cannot be configured. These are called **Kanban statuses**.

In addition, each team can divide the Kanban statuses into smaller stages, called **lanes**. Define lanes to reflect the work process that you want to follow in your team. For examples, see "Customize the storyboard" on page 135.

Benefits of using the storyboard

Using the storyboard to track a sprint's progress provides the following benefits:

- You can customize the storyboard to match the workflow that you have established in your team. The storyboard also provides guidance to the team about the stages and sequence a backlog item must follow.
- Purpose-designed graphs enable you to analyze the Kanban metrics collected by the storyboard. The graphs help you predict future performance, and fine tune your estimations and planning.
- Manage the workload, identify stale items and respond accordingly.

Kanban metrics

Using the storyboard helps you regulate the flow of backlog items through their work stages. The tool will also help you scope a sprint more effectively, and create backlog items that are more equal in size.

These goals are achieved through the following Kanban metrics provided by the storyboard:

Metric	Description
WIP Limit	Set limits on the number of items that can be in each lane and Kanban status. These limits are aimed at reducing bottlenecks, and help you balance the workload across the teams. If the number of backlog items in a lane or Kanban status goes above the limit, an alert will display.
Time Limit	Define how long it should take to work on items at different work stages. This setting is important to alert you on items that stay too long in any particular lane or Kanban status. There may be something blocking the story, or the backlog item may simply be too large.

Metric	Description
Cycle Time	Agile Manager calculates the cycle time for each item on the storyboard. The cycle time is the amount of time a backlog item spends on the storyboard between two specified statuses or lanes. Each team configures the boundaries of the cycle time. With this information you can compare the cycle times of different backlog items, draw conclusions about their sizes, and identify causes for longer cycle times.

What do you want to do?

Set the storyboard mode

The storyboard can operate in two different modes: **Scrum Mode** or **Kanban Mode**.

You can work with Scrum or Kanban methodologies, or use the storyboard's flexibility to manage a Scrumban methodology that suits your team best.

Scrum mode	Kanban mode
In Scrum Mode you continue to work in sprint cycles, but at the same time benefit from the flow and rules that the storyboard provides. At the beginning of each sprint you plan a sprint backlog, from which you take items for the storyboard. The sprint ends with a sprint closure, in which you reassign items that remain in the Planning and In Progress stages to the next sprint. The storyboard enables you to model the flow of work items through defined stages, and to gain insight into cycle times – the time it takes work items to complete their journey through the storyboard.	If you want to work on backlog items continuously throughout the release, choose the Kanban Mode. The flow of work items is governed only by the work-in-progress (WIP) limit and time limit set for each stage. **Tip:** Agile Manager requires you to define at least one sprint per release, so if you work with pure Kanban, define the release with a single sprint. You can also define multiple sprints, although this is not recommended for working in Kanban mode.

Set the storyboard mode

The storyboard mode is set at the release level, and applies to all the teams in a release.

1. In Configuration, select **Release** > **Release Details**.
2. Select the release you want to configure.
3. On the left sidebar, click **Additional Details**.
4. Under **Set Storyboard Mode**, select a mode.

After you change the storyboard mode, users must log out and log in again for the change to take affect.

Storyboard mode — detailed comparisons

The following table highlights some of the main differences between the two modes:

	Scrum mode	Kanban mode
Where do I take items from for the storyboard?	Drag items from the Sprint Backlog lane onto the storyboard, or add items from the release backlog. Adding items from the release backlog adds them to the sprint currently displayed.	
What is displayed on the storyboard?	Backlog items assigned to the current sprint.	Backlog items your team has worked on throughout the release. The Sprint Backlog lane displays only items assigned to the current sprint. When you move items into other lanes, they remain on the storyboard for the rest of the release.
What happens to open items at the end of a sprint?	On the Sprint Closure page, reassign any open items to the next sprint. Items assigned to previous sprints no longer appear on the storyboard.	Items remain on the storyboard for the entire release. If you are working with sprints, items in the Sprint Backlog column will change to the items assigned to the new sprint.

	Scrum mode	Kanban mode
How do I prepare the storyboard for the next sprint?	Create a sprint backlog that includes items you carried over from the previous sprint and other items from the release backlog.	If you are working in Kanban mode with sprints, create a sprint backlog that includes items you carried over from the previous sprint and other items from the release backlog. Items remain on the storyboard for the entire release, even after a sprint is completed. Items in the Sprint Backlog lane change with each sprint.

Customize the storyboard

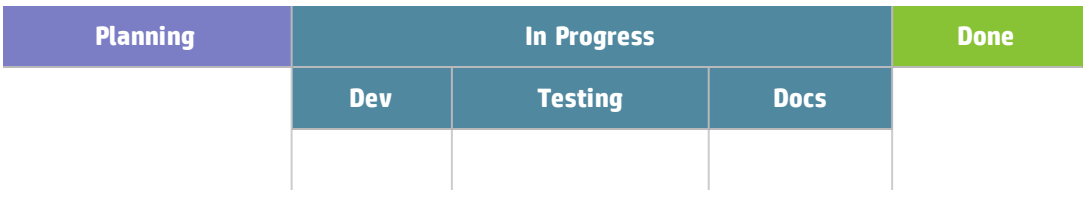
Design the storyboard according to your team's work process, and set storyboard rules.

Before learning about how to customize the storyboard, first read about the following concepts:

Storyboard lanes

Each team can customize the storyboard to match its own workflow. Each of the three predefined Kanban statuses can be split into lanes. Each lane represents a planning or development stage that the work items are required to pass through.

Example 1



Example 2

Planning			In Progress		Done	
Definition	Mockup	Reviewed	Dev	Testing	Docs	Done

Storyboard rules

For each of the Kanban statuses and lanes, set rules that control the flow of items across the storyboard:

- **Time Limit.** The amount of time an item can stay in a Kanban status or lane. The time limit is measured in days.

> **Note:** An item's stay time in a lane or Kanban status is measured continuously from the time it enters the lane or Kanban status. Therefore, the time limits you set should allow for time outside office hours.

- **Work in Progress (WIP) Limit.** The number of items that can be in a Kanban status or lane at the same time.

Cycle time measurement

Agile Manager tracks each backlog item's cycle time: the total amount of time an item spends on the storyboard. This information is useful for analyzing variances between work items, and comparing cycle times at different periods in the release.

You can configure whether the cycle time includes both the Planning and In Progress statuses, or just the In Progress status.

Status mapping

Each of the Kanban statuses on the storyboard is mapped to a backlog item status. When you shift items to a new status on the storyboard, its backlog item status changes accordingly, and vice versa.

In addition, for each lane that you add, you can specify the backlog item status that it corresponds to. The lane mapping overrides the Kanban status mapping.

The following table shows the mapping of Kanban statuses to backlog item statuses, provided that the Kanban status is not split into lanes.

Kanban Status	Backlog Item Status
Planning	New
In Progress	In Progress
Done	Done

How to customize the storyboard

1. In the configuration area, browse to a team details page for the team you want to configure.
   - For a workspace team, select **Workspace** > **Teams**.
   - For a feature team, browse to the release details page for the relevant release (**Workspace** > **Releases**). Select the **Teams** tab on the left of the release details page.
2. On the team details page, click the **Storyboard** tab on the left.
3. Click on any of the Kanban status columns.
4. In the **Edit Kanban Status** dialog box, set the overall **WIP limit** and **Time limit** for the Kanban status.

   Note: You cannot modify WIP or Time limits for the **Done** status.

5. Split the Kanban status into lanes. For each lane, define its **WIP limit**, **Time limit**, and corresponding backlog item status.

   Note: Removing existing lanes may affect data displayed in **Sprint** and **Release Control** charts on the Dashboard. For details, see "Sprint Control Chart" on page 72 and "Release Control Chart" on page 72.

6. In the **Planning** status, define whether the planning status will be included in the cycle time calculation.

Use the storyboard

Use the storyboard to manage the development life cycle of user stories and defects.

Note: Make sure the storyboard is customized for your team's needs. For details, see "Customize the storyboard" on page 135.

Your first visit

Open the **Release Management** > **Storyboard** page.

When you first enter the storyboard, the storyboard lanes may already contain several backlog items. This is because the storyboard statuses and lanes correspond to backlog item statuses. For example, if a backlog item was in the **In Progress** status, it will be positioned in the storyboard **In Progress** column. For details on the status mappings, see "Customize the storyboard" on page 135.

What do you want to do?

Add items to the storyboard

Use the following methods to pull items onto the storyboard:

- Drag items from the **Sprint Backlog** column into the first storyboard lane.
- Click **Add from Release Backlog**. The release backlog items are added to the storyboard and positioned in lanes based on their backlog item statuses. Backlog items you add to the storyboard from the release are added to the sprint backlog.

From the moment an item enters a lane or Kanban status, Agile Manager tracks its lane time, and compares it to the time limit.

Filter the storyboard

You have the following options to filter the storyboard:

- Use the filter box, located on the top left of the page.
- On the toolbar, select whether to display user stories, defects, or both.

Track cycle times

A backlog item's cycle time starts when the item enters the **Planning** or **In Progress** Kanban status, depending on the team storyboard's configuration. The cycle time ends when the item enters the **Done** Kanban status.

Rank items in a lane

You can change the rank of items within a lane. Drag the items up or down the lane.

Understand alerts

Alerts are given in two cases:

- When a backlog item exceeds the time limit for a lane or Kanban status.

  To view the up-to-date lane time, hover over the backlog item on the storyboard. The lane time is displayed in the tooltip.
- When a lane or Kanban status contains more backlog items than allowed by the WIP limit.

Watch items

Select backlog items you want to watch. You will receive mail notifications upon status changes to the items, or when they exceed a storyboard time limit.

Right-click a backlog item, and select **Add to My Watch List**.

Defect management

Report, manage, and track product defects.

Open the **Defect Management** page.

What do you want to do?

Report a new defect

Click **Add Item**. The Add New Defect dialog box opens.

The defects you create are added to the product backlog.

> **Note:** The following special characters are not supported in defect names: **\ ^ ***
>
> When using Internet Explorer, the **Attachments** field in the Add New Defect dialog box appears only in versions 10 and higher.

Import defects from a file

Import defects from a file into the product backlog or a release backlog.

On the **Product Backlog** > **Backlog** page or the **Release Management** page, click **More Actions** > **Import Backlog**, and browse to a prepared defect import file. Specify that you are importing defects, and Agile Manager validates the contents of your import file.

For import file templates and additional guidelines and field descriptions, see "Import backlog items" on page 25.

Associate defects with a feature

When you create a new user story or defect, you can associate it with a product feature. This enables you to group and filter user stories and defects by feature, and track the progress of the product at the feature level.

To associate existing items with a feature, select the items in the grid, and click **Set Feature**.

Plan for a release, sprint, or team

Select defects and click **Planning**  .

In the Planning dialog box, select a release, sprint, and team. You can specify a release together with a sprint or team, or both.

Appoint a defect owner

1. Select defects and click **Assign To**.
2. Select a user from the user list.

Assign defects to buckets

Assigning defects to buckets enables you to create categories of defects according to any criteria you choose.

1. In the buckets pane, click **Add Bucket**, and name the new bucket.
2. Select defects and drag them to the bucket. A bucket icon  is displayed in the Bucket column of the assigned defects. Hover over the icon to display which buckets a defect is assigned to.
3. To display the defects in a bucket, click the bucket name.

Manage the defect life cycle

Manage the defect life cycle by moving the defect through its statuses. The following defect statuses are available: **New**, **Open**, **Fixed**, **Closed**, **Propose Close**, **Deferred**, **Duplicate**, **Rejected**.

Defects have two statuses: the defect status and the corresponding backlog item status. These statuses are linked. When you change one status, the other status changes accordingly.

Change defect status

Defect Status	Backlog Item Status
New	New
Open	In Progress
Fixed	In Testing
Closed	Done
Propose Close	In Testing
Deferred	New
Duplicate	Done
Rejected	Done

Change backlog item status

Backlog Item Status	Defect Status
New	New
In Progress	Open
In Testing	Fixed
Done	Closed

Suggested uses for each defect status

Defect Status	Suggested use
New	This is the status provided when the defect is first opened. Leave this status unmodified until you are ready to start work on the defect.
Open	Use this status when you start to analyze or work on the defect.
Fixed	Use this status when you have committed a fix for the defect.
Closed	Use this status when the defect is tested and confirmed fixed.
Propose Close	Use this status if you are unsure whether the defect is completely closed, or if you would like to reject the defect, and need confirmation from others.
Deferred	Use this status to indicate that this defect will be fixed at a later date, and possibly not in the current release.
Duplicate	Use this status to indicate that another defect exists for this same issue elsewhere in the system.
Rejected	Use this status to indicate that the defect does not reflect an error in the system, and does not need to be fixed.

Watch defects

Add defects to your or to another user's watch list: right-click selected defects, and select **Add to My Watch List** or **Add Watch for User**.

Defects in your watch list are displayed in the My Watched Defects widget on the Dashboard.

In addition, you will receive mail notifications when watched items undergo certain changes. For details, see "Watch backlog items" on page 24.

Export defects to a file

Export the Defect Management grid to an Excel or CSV file to use defect data in other applications. Exported data includes any columns and filters displayed on the grid, and the **Description** and **Comments** fields.

1. Define a filter for the items you want to export and arrange columns you want to include.
2. Click **More Actions** > **Export Backlog to Excel** or **Export Backlog to CSV**.

Analyze defects

Use the following tools to analyze defects:

Quick widgets	Use the status bar graphs at the top of the page to gain quick insights into defect statuses and severities.	**Note:** The information in the quick widgets and the status bar refers to the defects in the current filter.
Status bar	Click a link in the status bar to view only defects at a certain level of severity or only defects assigned to you.	
Dashboard widgets	On the Agile Manager Dashboard, choose from a selection of defect widgets, or create custom graphs to track your defects. For more details, see "Dashboard and analysis tools" on page 65.	

Development monitoring

Monitor application development for a specific release or sprint, for specific features or themes, or completed by specific teams, using HP Application Lifecycle Intelligence (HP ALI).

HP ALI is a technology embedded in HP Agile Manager that can aggregate information from a variety of development tools, including open source tools as well as commercial ones. HP ALI establishes complete traceability, surfaces actionable information, and helps stakeholders in making informed decisions. HP ALI provides out of the box integration to source control management, build management/continuous integration, and IDE.

In Agile Manager:

- The **ALI Summary** page (**Release Management** > **ALI Summary**) shows high-level ALI metrics for the release
- The **Builds** and **Source Code** areas enable you to drill down to and compare detailed information about specific builds and commits

See also:

ALI development plug-ins

Use the HP ALI Dev plug-ins to connect to Agile Manager directly from your favorite IDE.

These plug-ins help developers connect to Agile Manager (and ALM) directly from their favorite IDEs.

Available for IDEA 12.* and 13.*

Able to install directly from IDEA (**Settings** > **Plugins** > **Browse Repositories** > **HP ALI**).

Open source (GitHub)

Available for Eclipse 3.6.x or higher (Eclipse Mylyn 3.10.x).

Download the zipped Eclipse here. (HP Passport required)

Available for Visual Studio 2010, 2012, and 2013.

Download the installation files here. (HP Passport required)

ALI user notifications

When ALI is configured for your release, Agile Manager can send email notifications if it detects that users may not be investing effort in the correct backlog items.

> For example, emails are sent if a user commits changes for a user story that is currently planned to a sprint other than the current sprint, or for a backlog item that is unassigned.

By default, notifications are sent to the user who committed the code, the user assigned to the backlog item, and the user who created the backlog item.

Individual users can modify settings to define the events they are notified about.

Modify ALI notification settings

1. In the configuration area, navigate to **Workspace** > **ALI Summary.**
2. Under **Notifications**, click **Configure Notifications**.
   - On the Notification Preferences page, select the rules you want to be notified for, or select the checkbox at the top of the table to select or clear all items.
   - Clear the **Receive e-mail notifications** option to unsubscribe from all ALI notifications.

Analyze build trends

Use the **Build Summary** page to evaluate build health in a release. For example, use the widgets to draw conclusions about the following questions:

- Are defects being worked on by severity?
- How is the development effort shared between defects and user stories?
- Identify unplanned work: what percentage of changes are not assigned to either user stories or defects?
- How stable are the builds?
- Is new code being tested?

> **Note:**
>
> All build information on the page is specific to a release and filtered by several parameters.
>
> - Make sure the correct release is selected.

- Select an application, build type or configuration, and time period.
- Determine whether you want to display failed builds and downstream builds. If you do not select downstream builds, only root builds are displayed.

Data aggregation: Metrics for parent builds display aggregated data of their downstream builds.

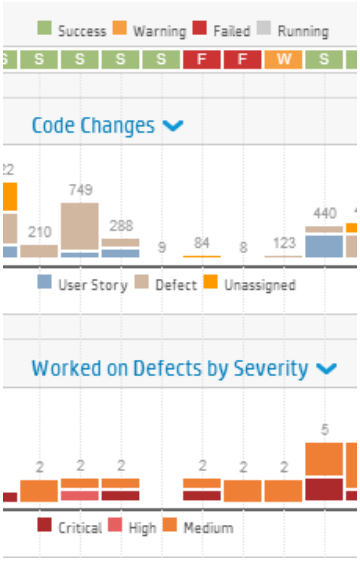
The **Build Summary** page displays the following widgets:

Success Rate and Average Build Time

An overall summary of the build success rate and the average build time.

Builds timeline

Select a build number to view detailed metrics. Drag the slider at the bottom of the timeline to view earlier or later builds.



- View the build status along the top row of the timeline  .
- Find correlations in builds between pairs of metrics. The metrics are displayed as bars on the timeline.

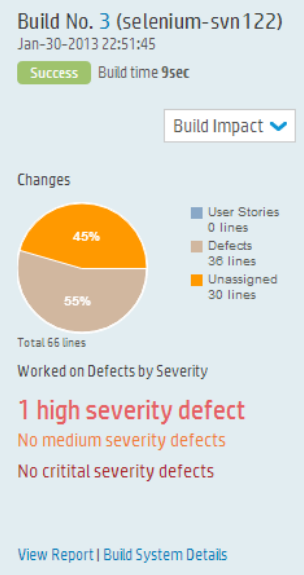
  Use the dropdown menus to select any two of the following metrics:

Code Changes	The number of lines of code changed in the build. The code changes are broken down by their association with user stories, defects, or neither.
Code Coverage	The percentage of code in the build that is covered by unit tests.
Test Success	The success rate of unit tests run on the build.
Worked on Defects by Severity	The number of defects worked on in the build, broken down by severity.

> **Examples of correlations**
>
> **Test Success - Code Coverage:** If the test success is stable, but code coverage is declining, the test success is not very informative.
>
> **Code Changes - Code Coverage:** If code changes are made but the code coverage is in decline, developers are not creating tests for their code.
>
> **Code Changes - Test Success:** If code changes are made but the test success rate is dropping, the new code may be breaking existing features.

Build details

View detailed information about a build.



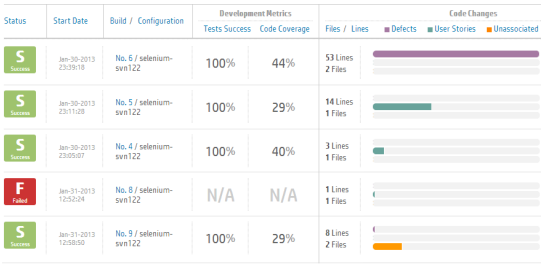
In the builds timeline, select a build and specific metrics to display.

More options:

View metrics for a range of builds	Select another build from the **Build Impact** dropdown to create a range. For details, see "Review a build range report" on page 148.
Display full build details	Click **View Report**. For details about the data shown here, see "View build report" on the next page.
Display build server statistics	Click **Build System Details**.

Builds grid

Compare key statistics for all builds, such as unit test success rate, code coverage percent, and number of files and lines affected by code changes in the build.



- Click a column header to sort the table.
- Click a build number to open a build report.

View build report

Access a build report to evaluate the quality of a build, and review the code changes connected to it.

To open a build report, click a build number. The widget bar along the top of the page displays summary metrics of the build.

> **Data aggregation:** Metrics for parent builds display aggregated data of their downstream builds.

The following widgets are shown across the page:

Changes	Defects	Development Metrics	Committers
Displays the number and percentage of lines of code that were changed in the build, grouped by their association with user stories, defects, or neither.	Lists the number of defects relevant to the build, grouped by whether they were newly detected, closed, or worked on in the build.	Displays success rates for tests run and percentage of code covered by unit tests in the build.	Lists the committers active in the build, in order of the percentage of code committed by each user.

Click a button below the widgets to display the following additional information for the build. When available, click **Show details** alongside a change set to view the change set details.

Delivered User Stories

Review the user stories associated with the build's code changes, displayed in a grid.

A list of change sets is displayed for each user story.

View by Defects

Review the defects associated with the code changes in the build, displayed in a grid.

A list of change sets is displayed for each defect.

Select one of the following options to view different types of defects:

- **Worked on Defects**
- **Detected Defects**
- **Closed Defects**

Change Log

A list of the change sets associated with the build, grouped by date.

Select **Show unassigned changes only** to filter out change logs for items already associated with a user story or defect.

Failed Tests

Details about failed unit tests, including details from the last commit, such as the committer and related backlog item.

Tree View

Review a build in context of its upstream and downstream builds. The selected build is displayed in bold.

- If the selected build is at the bottom of its tree, click **Show full hierarchy** to display the builds tree.
- Select the **Show Full Hierarchy** option to display the build's upstream builds.

Full details are displayed for downstream builds that are configured in ALI. To configure a build in ALI, click the **Add Configuration to ALI** link, and follow the on-screen instructions.

Show More

Select a build property to display:

Committers	Displays statistics of the users who committed changes to the build.
Build Artifacts	Lists the artifacts produced by the build. Click a link to download an artifact.
Build Notes	View and edit notes relating to the build.

Review a build range report

Review the impact of a range of builds using the **Build Impact** page. For example, review the stories delivered or defects fixed in a sprint build.

Do one of the following:

From the Build Summary page	Select a build in the timeline. In the **Build Details** widget on the right, select another build from the **Build Impact** selector to create a range.
From the build details page	Click the **Build Impact** link on the right. To modify the build number at the other end of the range, select another build from the build selector.

The **Build Impact** page displays the following widgets:

Changes	Defects	Committers
Displays the number and percentage of lines of code that were changed in the build, grouped by their association with user stories, defects, or neither.	Lists the number of defects relevant to the build, grouped by whether they were newly detected, closed, or worked on in the build.	Lists the committers active in the build, in order of the percentage of code committed by each user.

Click a button below the widgets to display the following additional information for the build range. When available, click Click **Show details** alongside a change set to view the change set details.

Delivered User Stories

Review the user stories associated with the code changes delivered in the range of builds, displayed in a grid.

A list of change sets is displayed for each user story.

View by Defects

Review the defects associated with the code changes delivered in the range of builds, displayed in a grid.

A list of change sets is displayed for each defect.

Select one of the following options to view different types of defects:

- **Worked on Defects**
- **Detected Defects**
- **Closed Defects**

Change Log

A list of the change sets associated with the range of builds, grouped by date.

Select **Show unassigned changes only** to filter out change logs for items already associated with a user story or defect.

Committers

Displays statistics of the users who committed changes to the build.

Analyze source code trends

Track source code changes and review their impact.

1. Open the **Source Code** page.
2. All source code information on the page is specific to the options selected at the top of the screen.
   a. Make sure the correct release is selected, and then select a time period within the release.
   b. Depending on the report you are viewing, filter information by application, team, and SCM branch as needed.
3. Review the overall **Development Effort**. This is expressed by the number of changed lines of code, broken down by their association with user stories, defects, or neither.
4. Analyze the source code using the following views:

Change Impact

Displays the user stories and defects impacted by code changes. The associated change sets are grouped under each user story and defect.

- Select whether to display code changes associated with user stories or defects.
- Filter the user stories and defects by application, team, and SCM branch.
- Expand the change sets below each user story or defect.
- Click a column header to sort the grid by that column.

Browse Code

- Displays a heatmap of the code repository, representing metrics such as defect density and code churn per folder.
- Enables you to explore and examine the repository at each folder level.

For more details, see "Browse the source code library" on the next page.

Change Log

Lists the change sets chronologically.

- Filter the change sets by team and SCM branch.
- To view only builds of a specific build configuration, select **Include Builds**, and then select the type of builds you want to view.
- Select **Show unassigned changes only** to list only changes that are not associated with user stories or defects.
- Click the path link to open the contents of the file in a separate tab.
- Click **[Diff]** to display a side-by-side comparison of the current file version with the previous version.
- Click **Show details** to display details of the change on the Source Code page.
- Click to display the Search box, and search the Change Log for items with specific keywords.the Search box to

Search the Change Log

Search for items that contains specific keywords in the commit message, names of files changed, or committer name. Search for names by SCM username, or the user's login name (email address).

To search, click to display the **Search** box, and enter the keyword you want to search for.

> **Note:**
>
> The search is performed by default in change sets from the last 31 days. To search older change sets, click **Search for Older Change Sets** in the search results.
>
> Searches are only supported when the **Include builds** option is cleared.

Browse the source code library

Analyze the quality of source code folders based on various metrics.

Click **Source Code > Browse Code**.

Learn More

The Browse Code page enables you to browse the repository's folder structure and display key metrics regarding the folders and files.

To ensure full functionality, make sure that the most recent version of the ALI plugin is installed on the build server. For details, see "Set up Hudson/Jenkins for ALI" on page 195 and "ALI Hudson/Jenkins plugin configuration" on page 196.

If the plugin is not configured correctly, the following folder and file data will not be available: Lines of Code, Relative Code Churn, Defect Density.

What do you want to do?

Analyze source code folders in a heatmap

The boxes on the heatmap represent folders that share the same parent folder.

The box sizes and colors represent the folders' values in one of the following parameters:

Size parameters

Parameter	Description
Lines of Code (LOC)	The total amount of lines of code included in an SCM folder.
Code Churn	The amount of added or changed lines of code in an SCM folder, between a baseline version and the current version of the files. The baseline version is taken at the start date of the specified time period.
Relative Code Churn	The code churn in an SCM folder, as a ratio of the folder's LOC.

Color parameters

Parameter	Description
Unit Tests	The success rate of unit tests run on an SCM folder.
Coverage	The percentage of code in an SCM folder covered by unit tests.
Defect Density	The number of defects associated with the code in an SCM folder, as a ratio of the folder's LOC.

To use the heatmap:

1. Set a **Time period**. The time period determines the range of data that is analyzed for the heatmap.
2. Select parameters from the **Size** and **Color** drop down lists.
3. Click a box to drill down to its child folders, or click a folder name in the grid below the heatmap.
4. Use the **Path** breadcrumbs under the heatmap to navigate to higher levels in the folder structure.

Analyze source code folders in a table format

The table below the heatmap displays the same folders as the heatmap. It also lists the files included in the same parent folder.

The following properties are displayed for each folder and file:

Property	Description
Unit Tests	The success rate of unit tests run on an SCM folder or file.
Defect Density	The number of defects associated with the code in an SCM folder or file, as a ratio of the folder or file's LOC.
Code Churn	The share of code changes in the folder or file associated with user stories, defects, or neither.

- Click a folder to drill down to its child folders and files.
- Click a file name to display its contents in a separate tab.
- Click **[Diff]** under a file name to compare the current contents of the file with an earlier revision.

Switch between online/offline browsing

Select **Online browsing** above the heat map to toggle between online and offline mode.

- When in online mode, the folder and file structure is loaded directly from the SCM branch.
- When in offline mode, the folder and file structure is based on the result of code parsing in the database.

Offline mode does not affect the metrics displayed: these are always loaded from the database.

When does Agile Manager switch to offline mode?

Code browsing will automatically enter offline mode in cases similar to the following scenarios:

- ALI DevBridge is down
- SCM repository is down
- Authentication error when connecting to SCM repository
- Network problem

When should I manually switch to offline mode?

Offline browsing is recommended if you have a slow connection to the SCM repository, or if you have a large folder structure that requires loading a larger amount of data.

View change set details

View details of a change set, and link it to work items.

Access a change set, and click **Show details**.

What do you want to do?

Change linkage to work items	Under **Linked work items**, review the items the change set is associated with. If the change set is linked to more than one item, scroll through them using the **Previous / Next Item** links. To change the linkage, use the following actions: **Link to User Story**, **Link to Defects**, **Remove Linkage**
View the contents of a file	The files included in the change set are listed in the grid on the bottom section of the page. Click a file name to open the contents of the file.
View the diff of a file	To display a side-by-side comparison of the current file version with the previous version, click the **Diff** link next to the file name.

ALI Q&A

How can I get a brief overview of what the development teams are doing and the development status of a release?

See **Release Management** > **ALI Summary**. You can see metrics which indicate problems, such as too many severe defects, unassigned changes, or frequently failing builds.

What was actually implemented/delivered during the last sprint?

In the **Source Code** tab, click **Change Impact**.You can see all user stories and defects that were changed. You can also link to any unassigned changes that may have been made.

Where can I review recent events to discover the reason for failing builds/tests?

In the **Source Code** tab, click **Change Log**. You can see the recent commits together with who committed which changes. You can also see which builds were executed together with the test results and coverage for each build.

Who has significantly contributed to a release?

See **Release Management** > **ALI Summary**. The Committers chart shows a list of developers sorted by percentage contribution to the release. The metric used to calculate the percentage is Lines Of Code (LOC).

Which developer can answer questions about a specific user story/defect?

Do the following:

1. Navigate to the User Story or Defect Detail page and click **Development Activity**.
2. Click **Active Developers** to see which developers worked on the user story/defect.

What is the development trend? Is the team fixing existing functionality or adding new functionality?

In the **Builds** tab, view the Build Summary trend chart. You can view a breakdown of the changes done in each build by user stories, defects, and unassigned.

What was delivered in a particular build?

To view the details of a specific build, click the **Builds** tab, and do one of the following:

- Select the required build in the timeline, and in the pane to the right of the timeline, click **View Report**.
- In the metrics grid, in the **Build / Configuration** column, click the required build.

In the **Build Detail** page, the **Delivered User Stories** tab and the **View by Defects** tab show you what was delivered.

What are the development metrics for a specific user story/defect?

Do the following:

1. Navigate to the User Story or Defect Detail and click **Development Activity**.
2. Click **Unit Tests** and **Code Coverage** to view the development metrics.

What was changed for a specific user story/defect?

Do the following:

1. Navigate to the User Story Detail page or to the Defect Detail page and click **Development Activity**.
2. Click **Commits Change Log** to view the changes that were made.

What changes were implemented last week?

To view changes that were implemented in a specific time period, do the following:

1. Click the **Source Code** tab.
2. Select the required **Time Period** from the drop-down list.

Can I filter the Build Summary page?

The following filters are available for the Build Summary page:

- Release
- Application
- Build category
- SCM branch

Additionally, select whether to include or exclude failed or downstream builds.

Can I filter the Source Code Summary page?

The following filters are available for the Source Code Summary page:

- Release
- Application
- Time period
- SCM branch

Additionally, filter the **Change Impact** grid to display user stories or defects.

On the **Change Log**, select to include assigned changes, or display only unassigned changes.

Tips and Tricks

Want to make working in Agile Manager better, quicker, more fun? We've collected some tips and tricks to help you out.

Navigation: Quick actions (CTRL+1)

Jump quickly from one area in Agile Manager all the way to the other side!
Press **CTRL+1** to open the **Quick Actions** dialog box.

Start typing the name of the tab you'd like to jump to, select it and press **ENTER**.

Agile Manager brings you straight there.

Navigation: Recent items

Not sure how to find an item you just had open?

Find it again in the **Recent Items** menu in the top-right corner of Agile Manager.

The **Recent Items** menu lists recently viewed:

- Individual backlog items
- Configuration tabs for specific releases and teams

Navigation: Open items in a new tab

Keep your original grid open in it's current display, and view individual items in new tabs.

Right-click an item in a grid, and select **View Details in New Tab**.

Refresh data to view changes made on other tabs.

Navigation: Links to themes and features in tooltips and Details View

Navigate to an item's theme or feature directly from grids or from the **Details** view.

In grids, hover over an item ID to display the item's tooltip. Navigate to the item's theme or feature using the links in the tooltip.

In the **Details** view, under **Release Plan**, hover over the **Theme** and **Feature** names to display tooltips with the link.

Content: Use backlog item templates

Create templates to use the same naming conventions, descriptions, tasks, and acceptance tests in similar backlog items.

Templates are saved per workspace, in the configuration area.

Select a single template as the default for a new user story or defect.

Add descriptions to your templates to help your users determine which one to use.

Content: Add multiple tasks or acceptance tests at once

There's no need to add each task or acceptance test separately.

Type them out all at once, and add numbers or bullets to each line.

Click **Save as <#>** tasks or **Save as <#>** acceptance tests instead of **Save**.

Content: Use workspace teams to retain assignments across releases

Do you ever need to push an item off to the next release, but know that the same team will be working on the item?

Use workspace teams for all your releases to retain the team assignment when you change an item's release.

This works whether you change the release using the **Planning** button, in a grid column, or in the **Details** view.

> **Note:** The workspace team must also be assigned to the target release.

Drag & Drop: Reorder tasks

Drag and drop tasks to reorder them, for example, in the order they should be performed.

In the **Details** view, or in the **Sprint Backlog** tab, drag a task up or down.

On the **Task Board**, drag tasks in any direction to place them where you want them.

Task order is retained throughout Agile Manager.

Drag & Drop: Add attachments

Add attachments to a backlog item as you create it!

In the Add New Item dialog box, just drag a file in to the Attachments field to add it to the new item.

For existing items, in the Details view, drag files to the Attachments field on the right.

Dashboard: Modify widget context

Modify dashboard context menu selections to display data for a different release, sprint, or team in contextual widgets.

Graphs configured to use contextual data will be automatically updated.

Configure widget settings to modify contextual values.

Suggestion: Use this function in status meetings that cover multiple teams, sprints, or releases.

Dashboard: Add custom graphs to the Gallery

If you like to create and reuse custom Dashboard graphs, add them to the gallery to find them again.

Create your custom graph, and click **Next** before clicking **Add to Dashboard**.

Save it to a private category to keep the graph just for yourself. You can find these again in the **My Widgets** section of the gallery.

Or, select a public category to make the graph available for others.

Drag & Drop: Drag individual tasks to team member buckets

Use the **Sprint Backlog** grid to assign tasks to users.

Ensure that the correct team is selected at the top to display the correct team members.

1. Select a backlog item in the grid to display its tasks in the pane below.
2. Select a task and drag it to a team member bucket on the right.

The assignment progress bars are automatically updated.

Grid actions: Multiple selections

Ever wanted to select many items in a grid at once? You don't have to select them one by one.

Use the **SHIFT** key to select multiple items at once. Select the item row, not the checkbox.

If you really have a lot of items, zoom out in your browser to select more at a time.

Grid actions: Use the column header options

Grids can be sorted and grouped by field.

If you have a column displayed for the field, you can sort and group from the column header.

Hover to the right of the column label, and click the arrow to display the dropdown menu.

Sort, subsort, or group grid items, or hide the column altogether.

Grid actions: Filter for empty numeric values

Sometimes you want to find all the items with an empty value for a specific field.

In the column header, enter two single quotes (**''**) to filter for empty values.

> **Note:** This works for numeric fields only.

Grid actions: Filter using footer items

Each grid has information displayed in the footer.

For example, the number of items selected, number of story points displayed, etc.

Bold and colored items are also links.

Click the bold, and red or green text to show only those items indicated.

Help Center: Find more details on any page in our Help Center

For details about a specific tab or dialog box, select **Help** > **Help on This Page**.

To learn more about Agile Manager concepts, browse directly to the Help Center home page (**Help** > **Help Center**).

Browse through the Help Center, or use search terms to find the data you're looking for.

Select a filter to search in specific area only.

Show or hide callouts for additional help inside Agile Manager.

Configuration

Perform administrative setup tasks for your site, workspaces, teams, and releases. In addition, configure your SCM and build servers to work with ALI.

To view configuration pages, you must be in the configuration area of Agile Manager. In the header, click the configuration icon.

> **Note:** Some configuration pages are only visible by Site or Workspace Administrators. For details, see "User roles and permissions" on page 42.

Workspaces enable you to apply scaled agile methodology in an enterprise, by providing an additional layer of data visibility. With workspaces, you can manage multiple projects or products on the same Agile Manager site, and apply common settings to all workspaces, such as custom fields.

Users can toggle between workspaces to which they are assigned. Access to items outside a user's workspaces is not strongly restricted.

The following table summarizes the settings that are defined at the **site** and **workspace** levels.

Site level	Workspace level
Workspaces: Manage the workspaces in your site. For details, see "Manage workspaces" on the next page. **Site users:** Add users from the LDAP system to the site. For details, see "Define site users" on page 165. **Custom fields:** Define user-defined fields that will be available in all workspaces. For details, see "Create custom fields" on page 162.	**Workspace users:** Add site users to the workspace. For details, see "Define workspace users" on page 173. **Workspace settings (e.g. estimation defaults, definition of done, notifications):** Define settings that will apply to the selected workspace. For details, see "Configure preferences and notifications" on page 169. **Applications:** Define applications that will exist in the selected workspace. For details, see "Define applications" on page 175. **ALI:** Configure ALI for the selected workspace. For details, see "Configure ALI integrations" on page 186.

This section includes:

For details about configuring links between Agile Manager and ALM, see t*he ALM Synchronizer for Agile Manager User Guide*.

Configure site settings

Site settings apply across the entire site, for all workspaces, and can be viewed and modified only by Site Administrators.

To configure site settings, click **Configuration** on the top right of the page, and select **Site** from the navigation menu on the left.

What do you want to do?

Manage workspaces

Tab: Site > Workspaces. Visible to **Site** Administrators only.

> **Note:** Each site supports a default of 100 workspaces. If you need more workspaces (maximum 400), contact your system administrator for assistance.

To define a new workspace, click **Add Workspace**.

To delete a workspace, select the workspace, and click **Delete Workspace**.

- You cannot delete a workspace that you are currently logged in to.
- If you delete a workspace while users are logged into the workspace, users may experience errors. Users will be required to log in again.
- You cannot delete the original, default workspace (the workspace with ID **1000**).

> **Caution:** Deleting a workspace deletes all releases, backlog items, and ALI data in the workspace!

Create custom fields

Tab: Site > Fields. Visible to **Site Administrators** only.

Create additional fields for the following item types:

User Stories, *Defects*, *Themes*, *Features*, and *Backlog Items* (which apply to both user stories and defects)

The new fields can be displayed in dashboard widgets, grids, filtered by, and used in the dimensions of the release planning board.

> **Example:**
>
> Create a custom **Single-Selection** field to indicate the reason a defect was closed, with values of **Fixed** and **Duplicate**.

You can also "Define required fields" on the next page.

Maximums for custom fields and field parameters

You can define up to 10 custom fields for each item type. These fields will be used by all workspaces and releases defined in the site.

The custom fields can contain the following data types: **Date**, **Free Text**, **List** (single or multiple selection), **Numeric**, or **Users**.

- The following special characters are not supported in custom field names: **\ ^ * #**
- When defining a **List** field, you can specify up to twenty list values.
- When defining values for a Multi Value list field, semi-colons (**;**) are not supported in list value names.
- Your 10 custom fields can include up to three **User List** fields.

Track custom fields over time

You can track **Single Value List** and **Numeric** fields over time, enabling users to add the new field to custom Agile graphs on the Dashboard.

Other custom field types are not supported in custom Agile graphs.

In the Add field or Edit field dialog box, select the **Track fields over time...** option.

- You can track three custom fields for each type.*
- Tracked data begins on the data that you enable tracking for the field, and users cannot display a custom field in Agile graphs unless it is being tracked.

Custom Summary graphs on the Dashboard can always display custom fields of the following types: **Date**, **Free Text**, **Single Value List**, **Numeric**, or **Users** . **Multi Value List** fields are not supported in custom Summary graphs.

For more details, see "Create a custom graph" on page 66.

Define required fields

Tab: Site > Fields. Visible to **Site Administrators** only.

Require users to define a value when first creating a user story, defect, theme, or feature. You can also "Create custom fields" on the previous page.

To define that a field is required:

1. Do one of the following:
   - Add a custom field
   - Select an existing system or custom field and click **Edit**
2. In the **Add Field** or **Edit Field** dialog box, select the **Mark this field as required** option.

> **Note:**
>
> - You cannot add, remove, or edit system fields, but you can define them as required.
> - Attachment fields cannot be marked as required.
> - To mark the **Application** field as required, use the **Workspace > Settings** configuration tab (available to Workspace Administrators).

Required fields are indicated with a red asterisk (*), in the **Add Item** dialog box. Additionally, required fields are indicated on the **Fields** configuration page in the **Required (in Add Item / Import)** and **Displayed in Add Dialog** columns.

- Field requirements are enforced *only* when adding items using the **Add Item** dialog box, importing backlog items, or converting a user story to a feature.

  They are not required when updating existing items in grids or on the Details page, when synchronizing items with ALM, or when adding items using the Agile Manager API.
- Some system fields are always displayed in the **Add Item** dialog box. This is indicated in the **Displayed in Add Dialog** column and cannot be modified.

  You can specify whether this field requires the user to define a value.

Maximum number of required fields

- A maximum of 15 fields can be displayed in the **Add Item** dialog box for each entity type, including out-of-the-box fields that are always displayed.

  Once this maximum is reached, you cannot define any more fields as required for that entity type.
- Setting a *Backlog Item* field as required counts towards the maximum for both user stories and defects.

Tip:

When adding or editing a field, a note is displayed indicating the number of fields already included in the Add Item dialog box.

To see which fields are included in the Add Item dialog box, do one of the following:

- Preview the dialog box as described below.
- Filter the **Fields** configuration page:

  - In the **Apply To** column, select the relevant entity type.

  - In the **Displayed in Add Dialog** column, select **Always displayed** and **Displayed when required**.

Preview the updated Add Item dialog box

On the Fields configuration tab, select a field and click **Preview** to view the Add Item dialog box for the relevant entity type, with all required fields included.

You may want to do this if you have many required fields and are concerned about crowding the Add Item dialog box for your users.

Caution: The preview dialog box includes your latest changes, but when you finish editing, you must log in again to see the changes in Agile Manager.

Importing items with required fields

If you import user stories or defects to Agile Manager, you must add additional columns to the import files for any fields marked as required, using the exact field names.

You may want to mark these extra columns as required by adding a comment to the column header, and save the file as a template for later use.

Note: When you import backlog items, field requirements for themes and features are not enforced. This enables you to import backlog items associated with new themes and features, which are created during the import.

For more details, see "Import backlog items" on page 25.

Define site users

Tab: Site > Users. Visible to **Site Administrators** only.

Site users include all users in the site, in all workspaces. View users' current login status in the **Logged In** column.

What would you like to do?

Add new user

Create a new user and include the user in the site.

Include users

Select existing users from the LDAP system to include in the site.

Display users by entering all or part of a **Full Name** or **Login Name** in the column headers.

This button is only enabled when LDAP authentication is configured for the system. If this button is disabled, use the " Add new user" above option instead.

Remove users

Remove the selected users from the site.

/ Activate/Deactivate users

Activate users to enable them to log in to the site.

Users you add or include in the site are automatically activated, provided there are available licenses.

You may want to note the users' last login date in the **Last Login** column. If you are juggling licenses between users, you can use this data to identify users who have not used Agile Manager recently. Deactivate these users to make room for other new users.

Note:

- If you do not see the **Last Login** column, use the column selector to add it to the grid.
- You cannot activate users if all licenses are already in use. Either purchase new licenses, or deactivate other users to make licenses available.

Assign users to roles

Assigns the selected users to specific roles.

To assign multiple users to the same roles, select the users and click **Assign to Roles**. The selected role is applied to all selected users.

> **Note:** When assigning roles to multiple users, you must select users that are assigned to the same workspace. If the users are assigned to multiple workspaces, first select the workspace where you want to make the change.

Each user can have one of the following roles:

Role	Description
Site Administrator	Has read and write access to **Site** configuration pages, the **Workspace** > **Users** configuration page, and the **Integrations** > **API** configuration page. **Note:** The default SA user created during the initial system configuration automatically receives the Site Administrator role.
Workspace Administrator	Has read and write access to all application pages and functions, as well as the **Workspace** configuration area.

Role	Description
Team Member	Has read and write access to all application pages and functions, with the following exceptions: • Workspace administrators can set permissions to prevent team members from deleting items created by others. When the **Allow Team Members to delete backlog items created by others** option is cleared, team members can only delete themes, features, and backlog items that they author. By default, this option is selected, and team members can delete any items. Regardless of configuration, Team Members can modify tasks and acceptance tests, regardless of who the author is. For details, see "Permissions" on page 171 configuration. • Team members have read-only access to the **Author** field. This field can be modified only by Workspace Administrators. • Team members have read-only access to public favorites on grid pages, to public Dashboard favorites, and to the public Dashboard gallery. Team members cannot create, update, or delete these items. Team members can also make only the following configuration changes: • Modify team work hours per day and working days in sprint for their team, from the bucket in the **Release Backlog** only. • Modify ALI configurations.
Viewer	Has read access only for all backlog items, grid pages, and the **Sprint Closure** page. Can also watch backlog items to receive notifications about status updates. Additionally: • Has read and write access to private favorites on grid pages. • Can create private buckets on the **Defect Management** page. • Has read access on the Dashboard, and read and write access for private Dashboard items and favorites.
Integration Administrator	Has read and write access to the **Integrations** > **Synchronizer** configuration page to configure NextGen Synchronizer.
Integration Bridge	Manages communication between Agile Manager and the NextGen Synchronizer Integration Bridge. **Note:** For security purposes, this user should have no other roles.

Set SCM User Mapping

Map the selected user to a specific code committer. This enables Agile Manager to associate code changes with a specific Agile Manager user.

User mapping also enables you to filter change impact reports and change longs by Agile Manager teams.

SCM users are mapped one at a time.

Assign to Workspace

Restrict the workspaces the selected users have access to. Users will be able to view only items associated with the workspaces to which they have access.

If you decide not to use multiple workspaces, all users are automatically assigned to the **Default** workspace.

Tip: If you have many workspaces, enter a search string to filter the workspaces displayed.

View license usage statistics

Tab: Site > Licenses. Visible only to **Site Administrators** on systems with concurrent licenses installed.

Displays the consumption rate of the concurrent licenses used on your site, including details about:

- The maximum number of available concurrent licenses;
- Peak license consumption points;
- The average consumption rate over time;
- Analysis tips based on actual license usage.

What do you want to do?

To modify the data displayed in the graph, select one of the following from the **Display** dropdown:

- Last Week
- Last Month
- Date Range (enter a date range)

To export the graph data to an Excel spreadsheet, click **Export**.

Configure workspace settings

Workspace settings apply only to the current workspace, and can be modified only by Workspace Administrators.

To configure workspace settings, click **Configuration** on the top right of the page, and select **Workspace** from the navigation menu on the left.

> **Tip:** If you are assigned to multiple workspaces, select the workspace you want to configure from the dropdown menu in the header (next to the configuration icon).

What do you want to do?

For details about the **Releases** and **Teams** tabs, see "Configure teams and releases" on page 175.

For details about the **ALI Summary**, **Builds**, and **Code** tabs, see "Configure ALI integrations" on page 186.

Configure preferences and notifications

Tab: Workspace > Settings. Visible to **Workspace Administrators** only.

Workspace preferences

Workspace preferences include the following settings:

Team member capacity

Define the default number of work hours per day team members are expected to put in. This, combined with the number of work days in a sprint, determines the team member capacity, and the amount of hours of tasks they can take on.

If needed, you can modify work hours for a specific team, or for each team member in each sprint. For details, see "Configure releases" on page 177 and "Manage teams for a release" on page 181.

> **Note:** Capacity settings are defined for all teams, including both workspace and release teams.

Estimated work

Define the number of story points that will be estimated by default for each new defect, and whether a new task is automatically created for new defects.

If you select to automatically create a task for each new defect, the new task is created using the text you enter in the **Default task description** field.

The **Default estimation for tasks** value is the number of hours that will be estimated by default for each new task. Use task time estimations to compare Planned vs. Actual effort.

For details, see "Use the Task Board" on page 123.

> **Note:** User stories do not have a default value. If you do not estimate a user story, its story points will remain blank.

Set required fields

Select this option to ensure that all **new** backlog items are associated with an application.

If you select this option, make sure you create the relevant applications. For details, see "Work with applications" on page 82.

Set a backlog item to Done

When you attempt to change a backlog item's status to "Done", Agile Manager can check whether all its acceptance tests have passed and all its linked defects are closed.

If ALI is configured for your release, Agile Manager can also check the code coverage and unit test success rates criteria. For details, see "Development monitoring" on page 142. These criteria will be enforced only for releases where ALI is configured, and only for stories or defects where metrics are available.

Toggle these parameters to skip or apply these checks.

> **Note:** Even if you are allowed to close backlog items with non-passed acceptance tests and open linked defects, a warning displays to inform you of the open items.

Weighted Shortest Job First

Use the WSJF method to help determine feature priorities. A feature's WSJF score is the ratio of its **Cost of Delay (CoD)** divided by its **Job Size**.

The **Cost of Delay** is the sum of the following components:

- Business Value
- Time Criticality
- Risk Reduction and Opportunity Enablement (RR|OE)

In Agile Manager, you can determine the weight of each of the CoD components.

To display the WSJF columns in Agile Manager, and to enable setting the CoD component weights, select **Enable WSJF**.

Permissions

Select or clear the **Allow Team Members to delete backlog items created by others** to define delete permissions for users assigned to the **Team Member** role.

When this setting is cleared, team members can only delete themes, features, and backlog items that they author.

For more details, see:

- "Define site users" on page 165
- "Define workspace users" on page 173

Workspace notifications

Define the events for which workspace users receive mail notifications.

Configure templates

Tab: Workspace > Templates. Visible to **Workspace Administrators** only.

Workspace administrators can design templates for creating new user stories or defects. A template defines a description, as well as tasks and acceptance tests, for each new item based on the selected template.

By default, new workspaces are defined with a default defect template, with a single task for fixing the defect, estimated at 6 hours. Modify or remove this template, or simply clear the **Set as Default Defect Template** option, as needed.

Create templates using one of the following options:

Add Template	Adds a new, blank template.
Duplicate Template	Creates a copy of a template that you can then modify. Click **User Stories** or **Defects**, and select the template you want to duplicate.

Import Template	Import a template from another workspace. **Note:** This option is only displayed if you are able to access multiple workspaces. 1. Click **Import Template**. 2. Select another workspace and the templates you want to import.

Modify templates by making changes, and clicking **Save** at the bottom of the screen.

Field	Description
Template Description	Define a description of the template, such as when to use the template. Users can view these descriptions when adding new items, in order to help them determine which template to use. In the **Add Item** dialog box used to create a new backlog item, hover over the icon displayed next to the template name to view the description. The icon is only displayed for templates configured with a description.
Set as Default	Set the current template to be used by default when creating new user stories or defects.
Story or Defect Name	Define a name for each new item created by the selected template.
User Story or Defect Description	Define a description for each new item created using the selected template.

Tip: Use the defined **Name** or **User Story/Defect Description** as a template or prefixes for names and descriptions in new items.

Users can add additional text to the **Name** field in the **Add Item** dialog box, or to the item name or description after creating the item.

Define workspace users

Tab: Workspace > Users. Visible to **Site Administrators** and **Workspace Administrators** only.

Workspace users are selected from site users, and are able to view only items in the workspace they are assigned to.

View users' current login status in the **Logged In** column.

What would you like to do?

Include Users

Add a site user to the current workspace.

In the filter box, enter all or part of a user name (**Full Name**) to filter the list of users.

Unassign Users

Unassign the selected users from the current workspace.

Assign to Roles

Assigns the selected users to specific roles.

To assign multiple users to the same roles, select the users and click **Assign to Roles**. The selected role is applied to all selected users.

> **Tip:** A solid blue square in the selection box indicates that some users are assigned to that role. Click the checkbox to assign that role to all of the selected users.

Each user can have one or more of the following roles:

Role	Description
Workspace Administrator	Has read and write access to all application pages and functions, as well as the **Workspace** configuration area.

Role	Description
Team Member	Has read and write access to all application pages and functions, with the following exceptions: • Workspace administrators can set permissions to prevent team members from deleting items created by others. When the **Allow Team Members to delete backlog items created by others** option is cleared, team members can only delete themes, features, and backlog items that they author. By default, this option is selected, and team members can delete any items. Regardless of configuration, Team Members can modify tasks and acceptance tests, regardless of who the author is. For details, see "Permissions" on page 171 configuration. • Team members have read-only access to the **Author** field. This field can be modified only by Workspace Administrators. • Team members have read-only access to public favorites on grid pages, to public Dashboard favorites, and to the public Dashboard gallery. Team members can not create, update, or delete these items. Team members can also make only the following configuration changes: • Modify team work hours per day and working days in sprint for their team, from the bucket in the **Release Backlog** only. • Modify ALI configurations.
Viewer	Has read access only for all backlog items, grid pages, and the **Sprint Closure** page. Can also watch backlog items to receive notifications about status updates. Additionally: • Has read and write access to private favorites on grid pages. • Can create private buckets on the **Defect Management** page. • Has read access on the Dashboard, and read and write access for private Dashboard items and favorites.
Integration Administrator	Has read and write access to the **Integrations** configuration area to configure NextGen Synchronizer.

Set Access to Applications

Restrict the applications the selected users have access to. Users will be able to view only items associated with the applications to which they have access.

To allow users access to items that have no application value, select the **(not set)** value. Clearing this value sets the Application field as required for any new items those users create.

> **Note:** Access settings take affect the next time the user logs in.

Assign to Team

Assign the selected users to a workspace team, if workspace teams are already configured on the **Workspace** > **Teams** configuration tab.

Workspace teams function across releases. If you want to assign the user to a **feature** team, you must do so within the context of the relevant release.

For details, see "Configure teams and releases" below.

Set SCM User Mapping

Map the selected user to a specific code committer. This enables Agile Manager to associate code changes with a specific Agile Manager user.

User mapping also enables you to filter change impact reports and change longs by Agile Manager teams.

SCM users are mapped one at a time.

Define applications

Tab: Workspace > Applications. Visible to **Workspace Administrators** only.

List the applications developed in your workspace. You can then associate features with applications, and plan backlog items according to their application.

For details, see "Work with applications" on page 82.

Configure teams and releases

Teams and releases are configured within a workspace. To configure team and release settings, click **Configuration** on the top right of the page, and select **Workspace** from the navigation menu on the left.

You can configure two types of teams: **workspace teams** to share across all releases in the workspace, and **feature teams** to be used within the scope of a single release only.

Workspace teams	Feature teams
• Reused in multiple releases. • Tab: **Workspace** > **Teams** • Administrators define a workspace team's settings, and users can track the team's performance across releases. • Backlog items are assigned to teams irrespective of the release assignment. This means that if you change a backlog item's release, and the item is assigned to a workspace team, the item retains the same team assignment.	• Created and used only in the scope of a single release. • Tab: **Workspace** > **Releases**. Click **Teams**, and then click a team name. • If you want to later reuse a release team in a different release, administrators can convert it to a workspace team, and then assign it to the other release. • Backlog items are assigned to teams within a specific release. This means that if you change a backlog item's release, and the backlog item is assigned to a feature team in the release, the team value is cleared.

Workspace teams and feature teams function the *same* in the following situations:

- When assigning backlog items to teams from the **Add Item** dialog box (when you first add the backlog item)
- When assigning backlog items to teams by using the **Team** column in grids
- When filtering grids or views by team

What do you want to do?

Manage workspace teams

Tab: Workspace > Teams. Visible to **Workspace Administrators** only.

Workspace teams are reused in multiple releases. Administrators define a workspace team's settings, and users can track the team's performance across releases.

Task	Description
Create a new workspace team	Click **Add Team**. A new team details page is opened for you to create a new workspace team. For details, see "Configure team settings" on page 183. This team will not be available in any releases until you modify the release assignments. For details about managing teams after creating a release, see "Manage teams for a release" on page 181.
Edit settings for an existing workspace team	Select an existing team and click **Edit Team** to modify the team settings. For details, see "Configure team settings" on page 183.
Delete an existing workspace team	Select an existing team and click **Delete Team** to completely remove it from the workspace. **Caution:** Related planning and reporting information is lost when you delete a workspace team.
Assign a workspace team to a release or multiple releases	Select an existing team and click **Release Assignment**. Select the release(s) you want the team to be available in. **Note:** When you assign a team to a release, you can also view team data for that release from the relevant release details page.

Note: You can also create **feature teams**, which are used only within the scope of a specific release.

Many of the team settings other than release assignments, such as team members, and storyboard settings, are identical for both feature and workspace teams.

For details, see "Manage teams for a release" on page 181.

Configure releases

Tab: Workspace > Releases. Visible to **Workspace Administrators** only.

Release settings apply only to the selected release, and can be modified only by Workspace Administrators.

- Filter the list of releases by any of the column headers.
- To create a new release, click ✚ **Create Release**.
- To view and modify a release's details, click the release name to open the release details page. For more details, see "Modify release settings" below.
- To directly modify release sprints, click the number of the sprint value. You are redirected to the Sprints tab for the selected release.

What do you want to do?

✕ Remove a release

To remove a release, on the release details page, click **Overview** > **Remove Release**. Any related planning and reporting information is also removed.

Duplicate release

You can create a new release based on the configuration of an existing release. The duplicated settings include: teams and team members (including both release and workspace teams), sprint duration, and sprint closure action items.

This is useful, for example, if the same teams carry on in the next release. You can tweak specific team and sprint settings later.

On the release details page, click **Overview** > **Duplicate Release**.

> **Note:** Duplicating a release duplicates its release teams. These are additional release teams available in the relevant release only.
>
> Duplicating a release also duplicates its workspace team settings, but the workspace teams themselves are not duplicated. Instead, they are simply assigned to the new release as well as previous releases.

Modify release settings

Do any of the following from the release details page.

Task	Description
Extend or shorten a release	Use either of the following options to extend or shorten a release: • Modify start and end dates. On the **Overview** tab on the left, under **Release Time Frame**, edit the **Start Date** or **End Date** fields. • Add sprints to or remove sprints from the release. On the **Sprints** tab on the left, add a sprint to or remove a sprint from the end of the release. The release end dates are automatically extended or shortened.
Modify team velocities and view release capacity	On the **Overview** tab on the left, modify team velocities in the **Release Expected Capacity** area. Watch the total expected capacity for the entire release aggregate in value in the bottom-right corner of the grid. This values is the total of all team and sprint expected capacities. **Note:** • When modifying team velocity for a workspace team on the release details page, team velocity is updated for the current release only. These settings do not affect other releases the workspace team is assigned to. • You can also modify team velocities on the team details page. For details, see "Manage teams for a release" on page 181.
Modify work days in the release	Release working days are the days of the week that are considered work days during the release. For example, if your company works Monday-Friday, select those days as working days. If your company has other working days, select them accordingly. On the **Additional Settings** tab, modify the selected **Release Working Days**.

Task	Description
Determine item reassignment behavior	By default, backlog items assigned to another team retain their current statuses. The new team continues work from the point that the original team left off. To change this behavior, on the **Additional Settings** tab, in the **Item Reassignment** area, select to reset the backlog item's status to New.
Set the storyboard mode	Determine the mode in which the storyboard operates. On the **Additional Settings** tab, select either **Scrum Mode** or **Kanban Mode**. For more details, see "Set the storyboard mode" on page 133.
Add attachments to the release	On the **Attachments** tab, click **Add Attachment** and select a document.

See also

"Configure release sprints" below

Configure release sprints

Tab: Workspace > Releases. Visible to **Workspace Administrators** only.

Click a release, and then click the **Sprints** tab on the left of the release details page.

What do you want to do?

Task	Description
Add a sprint to the release	Click **Add Sprint** or **Add Sprint to End of Release**. **Note:** Agile Manager allows you to create overlapping sprints, although this is not a recommended practice.
Move sprints	To move sprints forward or backward in a release, select the sprints you want to move, and click **Shift Sprints**. If, as a result of moving sprints forward, a sprint runs over the release end date, the release will be extended to include the final sprint. Alternatively, edit the sprint dates directly in the grid.

Task	Description
Remove sprints from the release	Select the sprint or sprints, and click **Remove from Release**.
Add a description of a sprint	The sprint name can also function as its description. Click the value in the **Name** column, and enter the new name. Keeping a sprint number makes it easier for users to understand the chronology of the sprints.
Assign teams to sprints	Click the value in the **Assigned Teams** column, and select the teams that will participate in the sprint. Alternatively, on the **Teams** tab, select the sprints in which each team will be available.

Manage teams for a release

Tab: Workspace > Releases. Visible to **Workspace Administrators** only.

Click a release, and then click the **Teams** tab on the left of the release details page.

The Teams tab shows data for any feature teams configured for the release, as well as any workspace teams assigned to the release. You can filter the teams displayed by using the search box at the top of the page.

> **Note:** Data for workspace teams is displayed only for the release you are currently viewing.

What do you want to do?

Task	Description
Assign an existing workspace team to the release (workspace teams only)	Click **Assign Workspace Team**, and select the workspace teams you want to include in the release. This option is enabled only when there are additional workspace teams defined for the workspace. To add teams to the workspace, do so on the **Teams** configuration tab. For details, see "Manage workspace teams" on page 176.

Task	Description
Create a new feature team (feature teams only)	Creates a new team for use in the current release only. Click **Create Feature Team**. A feature team cannot be used across other releases in the workspace unless you convert it to a workspace team. For details, see "Convert a feature team to a workspace team(feature teams only)" below.
Edit a team's and team member details (workspace and feature teams)	Click a team name, or select a team and click **Edit Team**. For details, see "Configure team settings" on the next page.
Delete a feature team (feature teams only)	Delete a feature team if you truly no longer need it in the current release. Select a feature team and click **Delete Feature Team**. When you delete a feature team, the **Team** field in any assigned backlog items is cleared. Individual assignments to team members (such as the **Owner** field) remain. To delete workspace teams, do so on the **Teams** configuration page. For details, see "Manage workspace teams" on page 176.
Unassign a workspace team from the release (workspace teams only)	Unassign a workspace team from the release if you no longer need it in the current release. **Note:** Unassigning a workspace team does not delete it from the workspace, and backlog items assigned to the team remain assigned to the team. Backlog items also remain assigned to the release. Team members working on those backlog items do so outside the scope of a release. Select a workspace team, and click **Unassign Workspace Team**.
Convert a feature team to a workspace team (feature teams only)	Convert a feature team to a workspace team if you want to reuse it across multiple releases. Select a feature team, and click **Convert to Workspace Team**. **Note:** Workspace teams cannot be converted to release teams.

Configure team settings

Configure settings for a specific workspace or feature team on the team details page. Team details pages are visible to **Workspace Administrators** only.

Access a team details page as follows:

Workspace teams	Feature teams
• From the Teams configuration page (**Workspace** > **Teams**) • Via a specific release (**Workspace** > **Release**; select a release, and click the **Teams** tab on the left).	• Via a specific release (**Workspace** > **Release**; select a release, and click the **Teams** tab on the left).

Team details are mostly similar for both workspace and feature teams.

> **Note:** When editing settings on a workspace team's details page, be aware that changes you make will affect all releases the workspace team is involved in.
>
> You know that you're editing a workspace team if the  is displayed under to the team name.

What do you want to do?

Delete a team (workspace or feature teams)

When you delete a team, team values in any assigned backlog items are cleared. Deleting a workspace team removes it from all assigned releases.

Click the **Overview** tab on the left of the team details page, and then click **Delete Team**.

> **Note:** You can also delete a workspace team from the **Workspace** > **Teams** configuration page, or a feature team from the release details page. For details, see "Delete an existing workspace team" on page 177 and "Delete a feature team (feature teams only)" on the previous page.

Convert a feature team to a workspace team (feature teams only)

If you want to reuse a feature team in multiple releases across the workspace, convert it to a workspace team.

When viewing the team details page for a feature team, click the **Overview** tab on the left, and then click **Convert Team**.

Converted teams function exactly the same as teams first created as workspace teams. All team data in the current release is preserved.

> **Note:** Workspace teams cannot be converted to feature teams.

Modify the description, team leader, and sprint velocity (workspace or feature teams)

Click the **Overview** tab on the left, and then modify any of the following:

Task	Description
Modify a team description	Use the free-text **Description** field to add details about the team.
Modify a team's default sprint velocity	Edit the **Default Sprint Velocity** value to estimate the number of story points the team will complete in a sprint. This helps determine the amount of backlog items the team can take on in a sprint. **Note:** You can modify this data per sprint, in the table below.
Define a team leader	Hover over the **Team Lead** value, and select a team leader from amongst the list of team members.

Modify release and sprint assignments (workspace or feature teams)

Workspace teams can participate in multiple releases across the workspace. For feature teams, you can define the sprints in which the team will participate.

Click the **Overview** tab on the left, and define releases and sprints in the **Release and Sprint Assignment** area.

Task	Description
Assign the team to a release (workspace teams only)	Click **Assign Team to Release**, and select the releases you want the team to participate in.

Task	Description
Select the sprints in which a team will participate (workspace or features teams)	• Workspace teams are limited to sprints in the releases the team is assigned to. Select a release from the **Assigned Releases** dropdown, and then select sprints from the **Available in Sprints** dropdown. **Note:** If you accessed a workspace team details page from a related release, you are limited to the release you browsed from. • Feature teams are limited to sprints in their release only. • View aggregated sprint data for the team in the grid below. To modify the expected velocity for a specific sprint, click the value in the **Expected Velocity** column in the grid.

Add and modify team members (workspace or feature teams)

Click the **Team Members** tab on the left, and do one of the following:

Task	Description
Add team members	Click **Add Team Members**, and select the workspace users you want to add to the team.
Define a team leader	Select a user, and then select **Team Leader** from the **Role** dropdown.
Edit workday hours	Edit the number of hours in a specific team member's workday. The default work hours per day (team member capacity) are set on the workspace level. For details, see "Team member capacity" on page 169.
Select sprint availability	Select the sprints in which the team member will be available. Available sprints only include those sprints in which the team is already defined as available. If you need to add a team member to a specific sprint, make sure to first add that sprint to the team availability.
Remove a team member	Select a user, and click **Remove Member**. Users that are removed from a team remain active in the workspace and can still be a member on other teams.

Tip: A user can be a member of several teams simultaneously. This is useful for a shared resource that contributes to different teams, such as a technical writer. In such cases:

- Set the user's **Work hours per day** on each team according to the share of the his/her time on each of the teams.
- Select the sprints in which the user will be available on each team.

Customize the team's storyboard (workspace or feature teams)

Click the **Storyboard** tab on the left and design the storyboard to match your team's work process.

For more details, see "Customize the storyboard" on page 135.

Analyze team data across releases (workspace teams only)

When viewing the team details page for a workspace team, click the **Analytics** tab on the left.

Team data is displayed for all releases the team is assigned to, such as expected and done story points, and average cycle times.

Hover over column headers for details about how each item is calculated.

Configure ALI integrations

Configure ALI in Agile Manager in order to work with the build and SCM servers in your local environment.

ALI is configured within a workspace. To configure ALI build and SCM servers, click **Configuration** on the top right of the page, and select **Workspace** from the navigation menu on the left.

Tabs: Workspace > ALI Summary, Builds, Code

Use the wizard to take you through each step of configuring ALI, or configure ALI manually.

Tip:

Configure one or more build jobs or source code branches, as needed.

You can configure multiple build jobs — all of the same type, or each of different types, such as some from Jenkins and some from TFS.

You can also configure multiple source code branches, either for multiple GIT branches, or for GIT, SVN, TFS, and so on.

For more details, see "ALI supported environments and frameworks" on page 190.

ALI configuration includes the following steps:

Integration

Install and configure agents that connect your build server and SCM repository with Agile Manager.

These agents are a set of applications that are required to push information from the servers to HP Agile Manager.

1. Download and configure agents for your build server. For details, see Hudson Jenkins Installation or "Set up TFS for ALI" on page 193.
2. Download and configure agents for your SCM server. For details, see SCM Agents.

Build Management

Add build server details and build server configurations.

Use the wizard to add a new build server or configuration or update an one. Click **Add Build Server** to do so manually.

Configure additional settings on the **Builds** page. For details, see:

- "Add and edit build servers" on page 198
- "Add and edit build configurations" on page 200

Source Code Management

- **Add repository and branch details.**

  Use the wizard to add a new repository, or click **Add SCM Repository** to do so manually.

  Configure additional settings on the **Code** page. For details, see:

  "Add and edit repositories" on page 204

  "Add and edit branches" on page 216
- **Set the commit pattern for each repository.**

  Commit patterns enable you to automatically connect code commits to the repository to work items.

  For details, see "Configure commit recognition patterns." on page 210

ALI validation

The ALI configuration is automatically validated on an hourly basis in the background, and users are notified about any errors found. This helps to discover any synchronization issues, such as expired passwords or connections that cannot be established.

Sharing ALI artifacts across workspaces

Site administrators can now enable ALI artifacts (build servers and SCM repositories) for use across multiple workspaces. This means that administrators from multiple workspaces can use the same build servers and SCM repositories when configuring ALI for the releases in their workspace(s).

Once an artifact is shared, most artifact configuration settings are read-only for workspace administrators. The exception is artifact credentials, which must be entered by the workspace administrator before the shared artifact can be used in the workspace.

For the build server, credentials are always the build server user name and password. For SCM repositories, credentials include user names, passwords, client certificates, and passphrases.

Share artifacts in the build server and SCM repository details pages. For details, see:

- "Add and edit build servers" on page 198
- "Add and edit repositories" on page 204

> **Note:** To remove sharing from an artifact, or to delete it, you must first verify that the artifact is not used in an ALI configuration in any of the site's workspaces.

ALI troubleshooting

How do I connect to a build server or SCM server with a certificate that is not signed by a trusted certification authority?

If your build or SCM server requires HTTPS protocol to be used to connect to it, the server's certificate (SSL Server Certificate) must be trusted by clients. If your server uses a self-signed certificate or a certificate signed by your custom certification authority, you must add your root or self-signed certificate to the trust store used by ALI.

You can use the Java keytool to add your certificate to the trust store:

```
keytool -importcert -alias my_custom_authority -trustcacerts –file cert_
authority.crt
```

This command adds the **cert_authority.cer** certificate to the trust store of your JRE/JDK. If your web server uses a different trust store, you need to specify it on the command line (-file).

How can I verify that a build server or SCM repository is configured correctly?

Do the following:

1. In Agile Manager, click **Validate Configuration** in the build server, build configuration, SCM repository, or SCM branch details pages.
2. Make sure that the build configurations and SCM branch configurations are linked to the correct releases. For SCM branches, verify also the date range that is specified for the release mapping. By default, the date range is defined by the release start and end dates, but the range can be customized. Note that only code changes that fall within the range are loaded.
3. In Agile Manager, click the **Source Code** or **Builds** tab, and select the **All Sprints** time period.

When trying to add a new build configuration to my server, what should I do if I cannot see my build job?

Make sure that the ALI build agent has been installed on the build server, and that ALI is enabled for your build job (ALI post-build action). If ALI is not enabled for the build job, the build configuration (job) does not appear in the list of build jobs available for selection on the build server.

It's taking a long time to load source code changes into Agile Manager. How long should I wait?

The initial load of source code changes may take several hours. It depends on the following:

- The size of your code base and its history. If you want to quickly verify the correctness of your setup, you can choose just the portion of your real code base (a sub directory), or you can specify an initial revision to load. Older revisions will be ignored.
- The responsiveness of your SCM server. If your server is busy, it will take longer to get information about changes.
- The connectivity from the bridge to your SCM server and also from the bridge to HP Agile Manager. If you use an HTTP proxy, its throughput and responsiveness play an important role.

How do I resolve an authentication problem of my build server – "HTTP Error 401 – Unauthorized"?

When a modified authentication mechanism is used by your Jenkins/Hudson server, you may need to turn on preemptive authentication. Do the following:

1. On the **Builds** page, click your server to open its details page.
2. Select the **Preemptive authentication** property, click **Edit**, and select the **Value** checkbox.
3. Verify that the user is permitted to obtain ALI build information. You can do this by logging in to Jenkins and verifying ALI links:
   - Top level **ALI Integration** link
   - Job level **ALI Integration** link
   - Build level **ALI Integration** , **Code Coverage**, **Test Results**, and **Code Changes** links

ALI supported environments and frameworks

This topic lists the environments and environment types that Agile Manager integrates with by default.

> **Note:** ALI is designed as extensible and provides APIs that enable you to add additional integrations. Use the ALI SDK to develop integrations that are not supported out-of-the-box. For details, see "ALI Extensibility" on page 223.

Build Server Systems

Jenkins: 1.532.1, 1.565.3

Hudson: 3.1.0, 3.2.1

Microsoft Team Foundation Server (TFS): 2012, 2013

> **Note:** Agile Manager works with all versions of build servers described above, but we recommend using the listed versions.

Agile Manager also provides plug-ins for build servers that automatically extract build information and metrics. Using the plug-ins, Agile Manager embeds the build information and metrics into your project.

Source Code Management (SCM) systems

> **Prerequisites:**

- The BASH version must be 4.0 or later to support pushing into multiple repositories.
- **For CVS repositories:** MS PowerShell 2.0 or later must be installed on Windows and enabled to run scripts.
- **For TFS repositories:** For full TFS system support, you must install TFS agent web services. One of the web services is designed to count the number of changed lines. If the service is not installed, the Changed Lines field always displays '0'.

SCM System	Recommended Versions	Tested on Versions
Subversion (SVN)	• 1.6 • 1.7	• 1.6.11 • 1.6.16 • 1.7.1 • 1.7.3
Concurrent Version Systems (CVS)	• 1.11 • 1.12	• 1.11.22 • 1.11.23 • 1.12.13
Microsoft Team Foundation Server (TFS)	• 2012 • 2013	• 2012 (11.0.51106) • 2013 (12.0.30723.0)
Perforce	• 2013.1 • 2014.2	• 2013.1/685046 • 2014.2/935585
Git	• 1.8 • 2.1	• 1.8.3.4 • 2.1.2

Note: Agile Manager works with all versions of SCM repositories described above, but we recommend using the listed versions will provide the best results.

The integration between Agile Manager and the source code repository is SCM client agnostic, meaning developers can commit changes from their current SCM clients (whether it is a command line utility or an IDE like Eclipse with the Tasktop plugin).

Unit Testing Frameworks

Agile Manager collects information from the following unit testing frameworks. The table lists the supported frameworks and with which build systems and versions these frameworks integrate.

Framework	Integrates with these build systems:	Recommended Build System Version
JUnit	• Jenkins • Hudson	
TestNG	• Jenkins • Hudson	• Hudson 0.8 • Jenkins 0.32
NUnit	• Jenkins • Hudson • TFS 2012 • TFS 2013	• Hudson 0.10 • Jenkins 0.14
Visual Studio Managed Unit Testing Framework	• TFS 2012 • TFS 2013	

Code Coverage Analysis Frameworks

Agile Manager collects information from the following code coverage frameworks. The table lists the supported frameworks and with which build systems and versions these frameworks integrate.

Framework	Integrates with these build systems:	Recommended Build System Version
Cobertura	• Jenkins • Hudson	• Hudson 1.1 • Jenkins 1.3
NCover	• Jenkins • Hudson	• Hudson 0.3 • Jenkins 0.3 (with NCover 3.4.18.6937 x86 trial version)
Visual Studio 2012 Code Coverage	• TFS 2012 • TFS 2013	

Build system integration

Agile Manager tracks information about builds together with their relationships to other Agile Manager entities in order to provide traceability between builds and release progress. Integration with a build server is a tool that allows you to measure the impact of code changes on software deliverables and build artifacts. Builds enable you to report what new code has been implemented and what the impact is on the release.

Agile Manager uploads information from the build server, including:

- Build artifacts (binaries produced by the build process)
- Content of the build (components, packages, files, etc.)
- New changes included in the build (what's new since the last build – change sets that make up the build).

With the uploaded information from the build server, Agile Manager links the build results, including information from tracking metrics installed on the build server, with work items. In doing so, Agile Manager is able to help you understand and track release progress.

This section includes:

Set up TFS for ALI

For full support of TFS 2012 or 2013 as the Build Server and as the SCM Repository, the HP ALI TFS Services must be installed on the TFS 2012 Server.

HP ALI TFS Services installation

You must install the `HP-ALI-TFS-Services_Installer.msi` file on the TFS server. The installer is part of the ALI bundle. This installer file provides the server environment with all the necessary validation. The HP ALI TFS installer is valid for both versions of TFS (2012 and 2013).

After installation, HP ALI exposes the following services:

TFS Service	Description
Line Count service	Counts changed lines within commits. **Using the Line Count service with TFS** In order to configure the Line Count service to work with TFS, you need to make additional modifications. The ALI SCM agent download for working with Microsoft TFS includes the Line Count service. The Line Count service is an extra ALI service used by the ALI-Microsoft TFS integration. This service allows ALI to get information about the number of modified lines within commits (check-ins). Without this service, ALI is unable to report on the number of changes performed. Therefore, it is strongly recommended to install this service. After installing the services, you must configure the location of the Line Count service file **(LineCountService.svc)**. In the Agile Manager Configuration area, navigate to the relevant SCM repository and configure the **Line count service URL** parameter. For details, see the **readme.txt** file included in the download.
Commit Message Validation service	Ensures that commit messages are aligned with the format configured for ALI on a specific SCM repository. **HP ALI Visual Studio extension (optional)** In addition to the basic commit message validation service, you can optionally install the HP ALI Visual Studio extension for Visual Studio 2012 and 2013 to validate commit messages. The extension must be enabled on the TFS project and installed on all developer machines being used to commit.
Build service	Exposes build information. **Build requirements for systems using TFS Servers** In order for your build to have full TFS support, your build job must meet the following criteria: • The build must have a Drop Folder defined. • The build must clean the workspace at the beginning of the build and all code changes must be checked out. • Results of the test and code coverage frameworks that were used must be stored as attachments in the `*.trx` and `*.coverage` files, respectively. Full TFS support enables you to get the test result and code coverage results provided by ALI for your builds. The TFS Build provider does not support the PUSH mechanism.

Set up Hudson/Jenkins for ALI

If you are working with a Hudson or Jenkins build server, install the following plugins on the build server to enable integration with ALI.

Hudson/Jenkins ALI agent

From the ALI Summary page, download the **Hudson** or **Jenkins Agent** plugin, and install it on the build server.

After installation, an **ALI Integration** link is added to the Hudson/Jenkins sidebar. Click the link to configure ALI plugin settings. For details, see "ALI Hudson/Jenkins plugin configuration" on the next page.

Force.com support

From the ALI Summary page, download the **Hudson** or **Jenkins Force.com** plugin, and install it on the build server.

Git/Perforce/TFS support

To add support for integrated SCM systems on the build server, follow these steps:

1. From the Hudson or Jenkins website, download and install the appropriate SCM plugin on the build server.

   An SCM plugin must be installed for the following SCM/build server integrations:

SCM	Build server
Git/Perforce/TFS	Hudson/Jenkins (all versions)
Subversion/CVS	Hudson 3 or higher

2. **Hudson 3 or higher:** From the Hudson website, download and install the **Maven** plugin.
3. From the ALI Summary page, under Available Build Agents, download the **Git**, **Perforce** or **TFS Support** plugin, and install it on the build server.

Test coverage information

To report build test coverage information in ALI, follow these steps:

1. Make sure one of the following plugins is installed and configured on the build server: **JaCoCo** or **Cobertura**.
2. **JaCoCo on Jenkins:** From the ALI Summary page, under Available Build Agents, download the **JaCoCo Support** plugin, and install it on the build server.

Hierarchical build support

ALI supports hierarchical builds on Hudson and Jenkins. ALI displays the build tree, and aggregated data of the downstream builds.

ALI supports the Hudson/Jenkins native downstream build trigger for hierarchical builds. In addition, the following plugins are supported on Hudson and Jenkins for hierarchical builds: **Build Flow** or **Multijob**.

For installation instructions, refer to the Hudson or Jenkins documentation.

ALI Hudson/Jenkins plugin configuration

The ALI Hudson plugin has a global configuration accessible from the global Configure System and the job scope configuration accessible from a specific job. For a detailed description, see the ALI integration plugin on the Hudson/Jenkins server, where every property is described.

Global Settings

Configure ALI settings that apply by default to all jobs. These properties can be overridden for a particular job.

1. Select **Manage Hudson/Jenkins > Configure System**.
2. Under the ALI Integration section, configure the following options:

Include the credentials in the SCM configuration	Specify if the username and password should be included in the SCM repository descriptor. If this security model is enabled, the user must also have "extended read" permissions for the credentials to be listed.  **Caution:** Be very careful when enabling this option as the credentials for the SCM repository associated with build configurations are exposed on REST endpoints as plain text.
Default HP AGM Server Connection	When selected, information about performed builds is sent to the Agile Manager server.

Job Settings

Configure ALI settings for specific jobs.

> **Note:**
>
> **Hierarchical builds.** If you maintain hierarchical builds, it is sufficient to configure the root builds only. All the builds will be displayed in ALI in a tree view, and the upstream builds will include aggregated data of the downstream builds.

1. Select a job and click **Configure**.
2. In the **Build** section, make sure the Analyze Source Code (ALI Integration) step is included in the build procedure. This collects LOC (lines of code) information from the repository.
3. In the **Post-build Actions** section, configure the following ALI options:

ALI Integration	You must select this to enable ALI integration for the job.
Test sources mapping pattern	This option enables you to determine test source locations based on actual test results. For details, including an example, see the ALI integration plugin help.
Update build information in HP AGM	This option sends information about performed builds is sent to HP Agile Manager immediately after a build is started, and again when it is finished.

4. Click **Advanced**, and configure the following additional options:

HP AGM Server Configuration	Overwrites the global ALI configuration properties, and updates the build information in HP Agile Manager for the build job.
NCover	Related to NCover code coverage for NET configurations. The *NCover report XMLs* specifies the generated raw XML report files, such as `myproject/target/coverage-reports/*.xml`. Basedir of the fileset is the root workspace.
Force.com	Related to Force.com integration. For details, see "Force.com integration" on page 219.

Clear LOC Cache

From the first build onwards, the LOC data is cached and used as a basis for the next build. Only files changed since the previous build are reanalyzed and merged with the cache. If you suspect that the data may be inaccurate due to problems with the build server, you can clear the cache, and force the server to gather the data again.

1. Select a job and click ALI Integration.
2. Under Line Of Code Metric, click the **Clear this cache** link.

Add and edit build servers

Add and edit build servers from the **ALI Summary** or **Builds** pages in the configuration area.

Tip: Add and edit build servers using the ALI Configuration wizard (recommended), or manually as described below.

In the wizard **Connect to Build Server** screen, select **Select existing build server** or **Create new build server**, depending on what you need to do.

1. ## Add and edit build servers manually

   Do one of the following:

   - To *add* a new build server, click **Add Build Server**.
   - To *edit* an existing build server's details, click the build server's name.

   The New Build Server or the Build Server details page opens.

2. ## Enter the server connection details.

   On the New Build Server or the Build Server details page, enter the connection details for the build server. Click **Validate Configuration** to check the connection.

   Note: The server location is a full path using the format `http://xx.xx.xxx.xxx/[yyyy]`. This address should be in DNS form. The `/yyyy` segment is optional.

   TFS users

   If you are using a TFS server, you must use the following format for the server address: `http://tfsServer:tfsport/tfsali/Service/BuildService.svc/ali/projectCollectionName`. The build service is part of HP ALI TFS Services which you have to download and install on the TFS server. The installer may be downloaded from the ALI Summary page

   Note: Elements of the TFS server address are defined as follows:

   - `tfsServer` is the address of the TFS server.
   - `:tfsport` is the port that the TFS server uses.
   - `/tfsali` is the name of the application on the IIS site defined during the installation of the HP ALI TFS Services.
   - `/projectCollectionName` is the name of the project collection.

   For details on ALI TFS Services, see "Set up TFS for ALI" on page 193

3. ## Share the build server.

   If you are a site administrator, and want to enable this build server to be used in other workspaces, click **Shared**. The button turns blue when the build server is shared. Click it again to disable sharing for this build server.

   > **Caution:** Only disable sharing after verifying that the build server is not used in ALI configurations in any workspace in the site.

   For details about shared artifacts, see "Sharing ALI artifacts across workspaces" on page 188.

4. ## Edit the build server properties.

   In the Properties section of the New Build Server or the Build Server details page, set the following server properties:

Property	Description
Retrieve incomplete data	Retrieves build information for all build configurations specified for the build server. Only those builds which have ALI enabled connect build information to the development information. By default, this property is disabled.
Preemptive authentication	Enables you to automatically enable communication between Agile Manager and the build server. By default, this property is enabled.

5. ## Add build configurations to a new server connection.

   a. Expand the **Select Configuration** section at the bottom of the page.

   b. Select the build configurations to add to the build server.

   All added configurations are displayed as sub-parts of the Build Server on the Builds page. For details on adding additional configurations, see "Add and edit build configurations" on the next page.

6. ## Edit the change detection settings for a build server.

   As part of its integration with the development process, Agile Manager detects new builds on the build server. In order to enable the retrieving of build information from the build server, you must set the **Change Detection** settings.

Set the following Change Detection settings in the Change Detection section of the Build Server details page:

Property	Description
Read changes from the Build Server and receive builds transmitted by Build Server Agents	Enables Agile Manager to automatically check the Build Server for new build information.
Change interval	The length of time Agile Manager waits to check the build server for new build information. By default, this interval is 60 minutes.
Synchronize now	Click this button to check the repository for new commits. To view the synchronization log, click **Show synchronization log**. For details, see "View the synchronization log" on page 219.

Add and edit build configurations

Add and edit build configurations from the **Builds** page in the configuration area.

Tip: Add and edit build configurations using the ALI Configuration wizard (recommended), or manually as described below.

In the wizard **Select Build Configuration (Job)** screen, select **Create New** or **Update Existing**, depending on what you need to do.

1. Add and edit build configurations manually. Do one of the following:

Task	Description
Add a build configuration	Select a build server in the grid, and click **Add Build Configuration**. Select an available build configuration from the list, and click **Save**. The Build Configuration Details page opens. **Note:** The list contains only configurations for Hudson/Jenkins jobs with ALI integration enabled.
Edit a build configuration	Expand a build server, and click a build configuration. The Build Configuration Details page opens.

2. Set the following build configuration properties:

Property	Description
Build Tracked [on/off]	If the build configuration is tracked, new builds from the build system are loaded into Agile Manager. Turn off this setting if the build configuration is obsolete, or is broken.
Source of Metrics [on/off]	The build configuration marked as the source of metrics is used to compute the statistics displayed for the associated release.
Name/Description	The name and description of the build configuration.
Release	The release with which the build configuration is associated. **Note:** This field is required.
Application	Select the application with which the build configuration is associated.
Build Category	Select the build category to which the build configuration belongs.
Source Code Resources	The source code repository associated with the build configuration on the build server.

Note: For details on enabling full support for build configurations on a TFS server, see "Build requirements for systems using TFS Servers" on page 194.

SCM system integration

Connect Agile Manager with your SCM server to enable Agile Manager to link code changes to user stories, defects, and tasks.

> **Tip:** To easily link your commits with Agile Manager, download the Tasktop plugin for Visual Studio, Eclipse or IntelliJ IDEA: https://hpln.hp.com/group/ide-ali. Installation instructions are included in the plugin download.

When you add or edit repositories and branches, define SCM policies that enable Agile Manager to link committed changes in the repository to development work items.

Enforcing SCM policies helps to ensure that developers follow prescribed guidelines and best practices. Team leaders can be confident that developers implement the right features and add required metadata to the committed change sets. Policy enforcement also provides invaluable help during stabilization periods as it is easy to ensure that developers fix only severe defects or to lock the code base of a release completely.

When you define a repository in Agile Manager, ALI recognizes a default pattern that it uses to associate code changes with work items.

This section includes:

Default commit patterns

When you commit code to the source code repository, include a message that includes details about the work items that the code changes relate to. Agile Manager parses the message and associates the code changes with the correct user stories, defects, and tasks.

This topic describes the default commit pattern that Agile Manager recognizes. To configure the commit pattern for a repository, see "Configure commit recognition patterns." on page 210

By default, Agile Manager recognizes the following commit pattern:

```
<default keyword> #id_1, #id_2, #id_3: <comment>
```

Commit message examples

```
fixing defect #12345, #54321: fix bug for UI

defects #6789: resolve issues on Core

user story #1001, #1002, #1003: fix caching and enhance functionality

implementing user stories #1004: finish US on release 1.0
```

Elements of the default commit pattern are described below:

- **<default keyword>.** Phrases that instruct Agile Manager to associate the code changes with user stories, defects, or tasks:

User stories	Defects	Tasks
user story	defect	task
implementing user story	fixing defect	implementing task
user stories	defects	tasks
implementing user stories	fixing defects	implementing tasks

- **Prefix.** Prefix ID numbers with a hash (**#**) sign.
- **ID number.** The user story, defect, or task number, as assigned by Agile Manager.
- **Separator.** Use a comma (**,**) to separate multiple items.
- **<comment>.** Free text describing the nature of the committed code.

SCM agents

The agent is a name for applications related to appropriate SCM systems required for PUSH mechanisms. The agent is a set of scripts or proprietary applications installed on a SCM server configured for listening on a SCM system. When you commit changes to a configured repository and branch, the agent checks the policies and pushes the change set to the Agile Manager server if the commit is allowed.

For more details on working with SCM agents, see "Edit the Change Detection settings." on page 209

Supported Operating Systems for SCM Agents

The SCM agents support deployment to the following operating systems:

- Red Hat Enterprise Linux 6.x (32bit, 64bit)
- SuSE Linux Enterprise 11.x (32bit, 64bit)
- Windows 2008 Server (32bit, 64bit)
- Windows 2008 R2 Server (64bit)

SCM Agent Download and Installation

To download the appropriate SCM agent, do the following:

1. In the Configuration view of HP Agile Manager, in the **ALI Summary** tab, under **Integration**, expand **Available SCM Agents**.
2. Browse to the correct build agent and download it to your SCM server.
3. Install the services. For details, see the **readme.txt** file included in the download.

Add and edit repositories

> **Tip:** Add and edit repositories using the ALI Configuration wizard while configuring a related build job (recommended), or manually as described below.
>
> If you are configuring repositories manually, it is still recommended to start by configuring a related build job, as described in "Add and edit build configurations" on page 200.

1. Click **Configuration** and either **Code** or **ALI Summary**.
2. Do one of the following:
   - To *add* a new repository, click **Add SCM Repository**.
   - To *edit* a repository, click the repository name.

   The New SCM Repository or repository details page opens.
3. **Add or edit the repository connection and authentication settings.**

   See one of the following tables, depending on your repository type:
   - "SVN Properties" on the next page
   - "CVS Properties" on the next page
   - "Git Properties" on page 207
   - "Perforce Properties" on page 207
   - "TFS Properties" on page 208

SVN Properties	Comments
Connection & Authentication details	Enter the server's connection details. The **Location** address must point to the actual root of the repository. To configure 2-way SSL (client certificate authentication), you must select a certificate and enter your encryption passphrase. **Tip:** If the URL for your SVN repository has an unknown root, you can find the root by entering **"svn info <URL>"** in the command line.
Diff template/ File template	The template for the address to use to access the diff link pages between revisions of a given code document or viewing the content of a given file. For details on diff links, see "Configure the links for an external repository viewer" on page 214. **Note:** If you are using Agile Manager's internal repository viewer, leave these properties blank.
Batch size	The number of revisions to include when uploading data from your SVN repository.
Initial revision	The number of the earliest revision to upload from your SVN repository.

After entering the connection information, click **Validate Configuration** to check the connection.

CVS Properties	Comments
Connection & Authentication details	Enter the server's connection details. The alias property must be the full path to the server, including the CVSROOT property, exactly as it is configured in your build system. For example, you could enter the CVSROOT property as `:pserver:username:password@host/cvsrepo`. The `CVSROOT` property is required for the `pserver` protocol.
CVS Protocol	The way of connecting to the CVS repository. **Note:** You must define the `pserver` protocol to begin uploading change sets.

CVS Properties	Comments
CVS Root	The address of the root directory for the CVS repository. You must use the following format: `[:protocol:][[<username>[:<password>]]][@<servername>][:<serverpath>]`
Commit interval	The maximum number of commits to include in a single change set uploaded to Agile Manager.
Initial date time	The earliest date from which to begin uploading commits from a CVS repository. **Note:** You must define the initial date and time to begin uploading change sets.
CVS Server Encoding	The character encoding used by the operating system on the computer on which the CVS repository is running, for example UTF-8.
Load interval	The number of days to include in a single grouping when uploading commits to Agile Manager from the repository.
Diff template/ File template	The template for the address to use to access the diff link pages between revisions of a given code document or viewing the content of a given file.. For details on diff links, see "Configure the links for an external repository viewer" on page 214. **Note:** If you are using Agile Manager's internal repository viewer, leave these properties blank.

After entering the connection information, click **Validate Configuration** to check the connection.

Git Properties	Comments
Connection & Authentication details	Enter the server's connection details. **Note:** • When entering the repository location, you must enter the entire Git URL. • To connect to your Git repository using HTTP/S, complete the **Username** and **Password** fields. • To connect to your Git repository using SSH authentication, select a **Security Key** option and complete the **Passphrase** field.
View diff link / View file link	The template for the address to use to access the diff link pages between revisions of a given code document or viewing the content of a given file.. For details on diff links, see "Configure the links for an external repository viewer" on page 214. **Note:** If you are using Agile Manager's internal repository viewer, leave these properties blank.
GitHub view diff/file link templates	When this property is selected, the GitHub web interface is used to view file diffs and files instead of Agile Manager's built-in repository viewer. **Note:** If you enable this property, you cannot enter a value for the **View diff link** and **View file link** properties.

After entering the connection information, click **Validate Configuration** to check the connection.

Perforce Properties	Comments
Connection & Authentication details	Enter the server's connection details. The location of the Perforce repository must include the host name and port of the Perforce server and the Depot name. You should use the format `host:port//depot_name` for the location.

Perforce Properties	Comments
View diff link / View file link	The template for the address to use to access the diff link pages between revisions of a given code document or viewing the content of a given file.. For details on diff links, see "Configure the links for an external repository viewer" on page 214. **Note:** If you are using Agile Manager's internal repository viewer, leave these properties blank.
Batch size	The number of revisions to include in a single upload group when uploading data from your Perforce repository.
Initial revision	The number of the earliest revision to upload from your Perforce repository.
Charset name	The Charset name used for communicating with the server in Unicode mode. **Note:** This property is required when communicating with the server in Unicode mode. The property must be empty when communicating with the server in non-Unicode mode.

After entering the connection information, click **Validate Configuration** to check the connection.

TFS Properties	Comments
Connection & Authentication details	Enter the server's connection details. The location of the repository must include the host name and port of the TFS Server and the name of the Project Collection. You should use the format `http://tfsServer:tfsport/tfs/ProjectCollectionName`.
View diff link	The template for the address to use to access the diff link pages between revisions of a given code document or viewing the content of a given file.. For details on diff links, see "Configure the links for an external repository viewer" on page 214. **Note:** If you are using Agile Manager's internal repository viewer, leave these properties blank.
Batch size	The number of revisions to include when uploading data from your Perforce repository.

TFS Properties	Comments
Line count service URL	The URL for count line service for your TFS server. **Note:** You must install ALI TFS Services to use this the Line Count service. This service is part of HP ALI TFS Services which you have to download and install on the TFS server. The installer may be downloaded from the ALI Summary page. For details, see "Set up TFS for ALI" on page 193. You only need to provide the property value if the line count service URL for your TFS server is different than the default value (`http://tfsServer:tfsport/tfsali/Service/LineCountService.svc`). If the URL is different from this default, and you do not provide the URL, then loaded code changes for all uploads report 0 changed lines and the branch is created with warnings.
Initial revision	The number of the earliest revision to upload from your TFS repository

After entering the connection information, click **Validate Configuration** to check the connection.

4. **Share the repository.**

   If you are a site administrator, and want to enable this repository to be used in other workspaces, click **Shared**. The button turns blue when the repository is shared. Click it again to disable sharing for this repository .

   Caution: Only disable sharing after verifying that the repository is not used in ALI configurations in any workspace in the site.

   For details about shared artifacts, see "Sharing ALI artifacts across workspaces" on page 188.

5. **Edit the Change Detection settings.**

   As part of its integration with the development process, Agile Manager detects new builds in the source code repository. In order to enable the retrieving of repository information from the repository, you must set the Change Detection settings.

Set the following Change Detection settings under **Change Detection**, on the SCM Repository details page:

Property	Description
Read changes from the SCM Repository and receive builds transmitted by SCM Repository Agents	Enables Agile Manager to automatically check the repository for new build information.
Change interval	The length of time Agile Manager waits to check the repository for new commit information. By default, this interval is 60 minutes.
Synchronize now	Click this button to check the repository for new commits.

> **Note:** If you use the wizard, select to synchronize immediately upon completing the wizard, or leave the repository to synchronize at the next defined change interval.

To view the synchronization log, click **Show synchronization log**. For details, see "View the synchronization log" on page 219.

6. **Configure commit recognition patterns.**

   Detecting and maintaining traceability between change sets and work items is based on a commit message that the developer provides when committing a change to the SCM repository. Agile Manager enables you to customize settings that help Agile Manager link the changes included in the change set to work items.

   You configure the commit pattern settings when adding a new SCM repository or editing existing an existing one.

   For details on the default commit pattern, see "Default commit patterns" on page 202.

   Do one of the following:

   - "Associate code changes with alternative identifiers" below
   - "Configure the commit pattern" on page 212
   - "Use an editor to configure the commit pattern" on page 213

   Associate code changes with alternative identifiers

   By default, commits are associated with user stories or defects based on their Agile Manager IDs. If you maintain another set of identifiers for your user stories or defects, developers can commit changes with reference to these alternative identifiers. For example, if you synchronize your workspace with an external defect tracking tool, such as ALM, in their commit messages, developers can reference the identifiers generated by that tool.

The alternative identifiers are stored in Agile Manager in a custom field. The commit message needs to include a phrase indicating that Agile Manager should look up the identifiers in the custom field. For example: `Resolve Jira issue`.

a. In the Repository Details page, expand the **Commit Pattern** area. Make sure the **Basic** view is selected.

b. In the Defects or User Stories section, add a keyword or phrase that indicates that the commit is using an alternative identifier.

c. In the **Field** list, select the custom field that stores the alternative identifiers. Agile Manager will look up this field if it identifies the keyword you typed in the commit message.

Configure the commit pattern

a. In the Repository Details page, expand the **Commit Pattern** area. Make sure the **Basic** view is selected.

b. Set the following parameters:

Parameter	Description
Keywords	Edit the list of keywords that Agile Manager uses to link a commit to a defect, user story, or task. If any of the words in the list are included in the commit message, the code changes are automatically associated with a defect, user story, or task. You can specify the field in which Agile Manager looks up identifiers. See "Associate code changes with alternative identifiers" on page 210 above.
ID prefix	The character or text that precedes work item numbers in the commit message. **Examples:** #, $, NUM
Include default Tasktop commit pattern	Generates default commit messages in the developer's repository commit tool when the repository commit tool has the Tasktop plugin installed. This option is enabled by default.
Case sensitive commit message	Instructs Agile Manager to read the commit message as case-sensitive. This option is disabled by default.
Multiple Defects, User Stories, or Tasks are separated by	Specify the character that separates multiple defect, user story, or task numbers in the commit message.
Keywords are located [At the beginning / Anywhere in the message]	Specify where Agile Manager searches for the specified keywords in the commit messages.
User commit message is separated by	Specify the character that indicates that the developer's comments follow.

Use an editor to configure the commit pattern

a. In the Repository Details page, expand the **Commit Pattern** area.

b. Select the **Advanced** view.

The text editor includes code converted from the settings in the Basic view.

c. Edit the commit pattern code. Use the following examples as guides.

"Example 1" below

"Example 2" below

"Example 3" below

Example 1

Pattern:

```
([fixing] REGEX('defects?') IDLIST(DEFECT) | [implementing] REGEX
('user stories?') IDLIST(REQ) ) : TEXT
```

Sample commit message:

```
"fixing defect #56721: something really serious was fixed"

"defects #57893,#61432: division by zero"

"user story #1: domains"
```

Example 2

Pattern:

```
(UNTIL(RE '((BUG)|(REQ))#') (IDLIST(DEFECT lead='((BUG)?#)?' sep=',' )
| IDLIST(REQ lead='((REQ)?#)?' sep=',' ) ) ){0,} [TEXT]
```

Sample commit message:

```
"This commit fixes BUG#1,#2 and implements REQ#4,REQ#5 making the
product faster (resolving BUG#7)."
```

This pattern matches all inputs and extracts of any found 'BUG#' and 'REQ#' patterns. Such an open pattern may not be suitable for enforcing common policy, but it can be useful when data from legacy repositories are loaded in the "read-only" mode, for example, for reporting purposes.

Example 3

Tasktop Pattern:

```
(REGEX('.*? - task DEF') IDLIST(DEFECT lead='' sep='') | REGEX('.*? -
task REQ') IDLIST(REQ lead='' sep='') | REGEX('.*? - task TASK')
IDLIST(TASK lead='' sep='')) : TEXT
```

Matches default Tasktop messages:

```
"OPEN - task DEF10: http://host:9090/qcbin;DEFAULT;ALI_DEV-DEF10"

"Incomplete - task REQ42: http://host:9090/qcbin;DEFAULT;ALI_DEV-
REQ42"
```

d. Use the following options to configure the advanced commit patterns:

Option	Description
Test Against Existing Commits	Enables you to test the advanced commit pattern text against settings already entered for existing repository commits.
Case Sensitive commit messages	Enforces case sensitivity on the advanced commit text.
Restore Defaults	Reverts your changes and replaces the default keywords as they were displayed before your changes.
Test	Enables you to test a custom message for correct syntax before entering it into the commit pattern. Enter your text into the edit field at the bottom of the tab and click **Test**.

Configure the links for an external repository viewer

As part of connecting changes in your code to the development work process, Agile Manager provides a built-in repository viewer for viewing the content of a specific file or the differences between versions of a file. If you are using Agile Manager's internal repository viewer, you do not need to configure the **Diff link** and **File link** properties in the Repository Details page.

You can use an external repository viewer, such as ViewVC, to view files and version differences. Enter the template for the links to the file content and diff views:

1. Do one of the following:
   - Provide the template of the HTTP link to the file view of a selected file
   - Provide the template of the HTTP link to the diff view of a versions of a selected file.
2. **Include the built-in variables as part of the template (if needed).**

   Agile Manager contains a number of tags to use in your template:

Tag	Description
`${filePath}`	The path to the file within the repository.
`${revision}`	The revision number of the selected file.
`${fromRevision}` (diff links only)	The previous revision number to use in the file difference view.
`${fromFilePath}` (diff links only)	Used if the selected file was moved from its original location after the original revision but before viewing the diff link.

TFS Server (2012 and 2013) Repository Users

To view the file stored in a TFS Server repository, use the following template:

```
http://tfshost:tfsport/tfs/_COLLECTION_/_versionControl/changesets#cs=${'revision'}'&path=${'filePath'}'&version=${'revision'}'&_a=contents
```

To view the difference between versions of a file, use the following template:

```
http://tfshost:tfsport/tfs/_COLLECTION_/_versionControl/changesets#opath=${'fromFilePath'}'&oversion=${'fromRevision'}'&mpath=${'filePath'}'&mversion=${'revision'}'&_a=compare
```

The TFS link template includes the following tags:

Tag	Description
TFSHost:TFSPort	The address of the TFS server.
COLLECTION	The name of the collection where the requested file's project is a member.
path (file links only)	The full path to the source file, including the $ symbol and project name). **Example:** `$/ProjectName/folder/readme.txt`
cs, version (file links only)	The TFS change set ID.

opath (diff links only)	The full path to the original source file, including the $ symbol and project name). **Example:** `$alireplica/alik/pom.xml`
oversion (diff links only)	The TFS change set ID of the original revision of the selected.
mpath (diff links only)	The full path to the modified source file, including the $ symbol and project name). **Example:** `$alireplica/alik/pom.xml`
mversion (diff links only)	The TFS change set ID of the modified file.

Git Repository Users

The **Git** template for file links and diff links includes the following tags:

Tag	Description
`${blobId}`	The blob SHA.
`${commitId}`	The commit SHA.
`${branchName}`	The name of the branch.
`${filePath}`	The file path within the Git repository (without the leading **/**).
`${prevBlobId}` (diff links only)	The blob SHA for previous version of file.
`${prevCommitId}` (diff links only)	The commit SHA for the parent commit.
`${prevFilePath}` (diff links only)	The previous file path within the Git repository (without the leading **/**).

Add and edit branches

Tip: Add and edit branches using the ALI Configuration wizard while configuring a related build job (recommended), or manually as described below.

If you are configuring branches manually, it is still recommended to start by configuring a related build job, as described in "Add and edit build configurations" on page 200.

1. Click **Configuration** > **Product** > **Code**.
2. Do one of the following:
   - To *add* a new branch: Select a repository, and click **Add SCM Branch**.
   - To *edit* a branch: Expand a repository, and click a branch.
3. On the SCM Branch page, set the following branch properties:

Path

Perforce: Set the branch path without the Depot name. For example, if the branch is located at `//depot/HelloWorld/releases/release-1.0/...` then the path should be `/HelloWorld/releases/release-1.0`.

TFS: Use the branch path to the project path without a leading `$`. For example, if the project is located at `$/TestApp`, then the branch path should be `/TestApp`.

> **Note:** Branch paths can contain sub folders.

Git: Always set the branch path to `/`.The field name should be set to the real Git branch name.

> **Note:** After entering a value, click **Validate Configuration** to test the link.

Branch

CVS: A branch name only has meaning for some CVS repositories. Do not fill this property for SVN.

Perforce: Do not use this field even if the branch is named.

Git: Use only the simple branch name in the format `master`, and not `refs/head/master`.

> **Note:** After entering a value, click **Validate Configuration** to test the link.

Last Change Read

CVS: The date-time stamp for which the change sets are read.

SVN: The last revision number for which change sets are read.

GIT: The commit hash (revision), space, and branch name.

> **For example**

Master	`00988ff4a833eea670feab6bcbcf5ed93f9add77 master`
A branch named **2.10**	`a12a5d41b0fba9934e9569cbae87e61599373ea2 2.10`

Note: Populate this field when creating a new branch to read changes only starting from branch creation. Leaving this field blank will include all changes starting from the beginning of the master/HEAD branch.

Check-in Policies

- **Commit message must match defined pattern.** Causes commits for which the commit message doesn't match the predefined pattern to be refused by the agent.
- **Change Set refers to a user story.** Enforces that every commit must refer to a user story. Optionally, set the priority levels user stories must have to allow users to check in associated changes.
- **Change Set refers to a defect.** Enforces that every commit must refer to a defect. Optionally, set the severity levels defects must have to allow users to check in associated changes.
- **Add this note to the System message when Commit is blocked.** Sends custom system messages that inform a user when a commit is blocked with the applied note.

Note:

- Check-in policies require the installation of an agent in order to function. For details, see "SCM agents" on page 203.
- Check-in policies do not work on Git repositories.

Locking Policies

Disallow commits except for the following. disallows all the commits coming to the given branch other than the following exceptions:

A list of users allowed to commit to the branch can be specified though the branch is locked by applying the user name(s) of SCM users who are permitted to commit.

A list of defects specified for which committing changes are allowed is created by clicking Add and providing the ID of the defect. Defects are removed from the table by clicking Remove.

Add this note to the System message when Commit is blocked. Sends custom system messages that informs a user when a commit is blocked with the applied note.

> **Note:**
>
> - Locking policies require the installation of an agent in order to function. For details, see "SCM agents" on page 203.
> - Locking policies do not work on Git repositories.

Release

You can associate a branch with multiple releases. Click **Add** and select the releases.

The start and end dates of the releases are taken from the release settings. You can change these manually in the grid.

Change sets from the branch are associated with a release during the time period specified for that release.

Application

Select the application with which the branch is associated.

View the synchronization log

Access ALI synchronization logs from the build server SCM repository configuration page. In the Change Detection area, click **Show synchronization log**.

ALI synchronization logs display details such as when the last synchronization started, and the user who started the task. If the task has completed, the log also shows the time that the synchronization finished.

The page shows the most recent 200 log records. Click **Load previous...** to view earlier records.

Click **Refresh** to refresh the data displayed.

For details, see "Add and edit build servers" on page 198 and "Add and edit repositories" on page 204.

Force.com integration

This integration allows teams developing for the Force.com platform to benefit from all the features that HP ALI brings to standard development. Although all the source code is stored, compiled, and tested in the Cloud, ALI establishes traceability between code, work items (user stories and defects), and builds.

> **Note:** Force.com integration was tested on Force.com version API 22.0.

This section includes:

Prerequisites for Force.com and ALI

The integration between Force.com and HP ALI requires:

Requirement	Details
Storing Force.com source code in an SCM system	For the complete list of SCM systems supported by HP ALI see "ALI supported environments and frameworks" on page 190.
Hudson or Jenkins	• Hudson/Jenkins plugin supporting use of SCM (SVN and CVS are supported by default). • HP ALI Hudson/Jenkins plugins (download from the ALI Summary page). • HP ALI Hudson/Jenkins Force.com plugins (download from the ALI Summary page). For details, see "Set up Hudson/Jenkins for ALI" on page 195.
A build management server	Configure a build management server to deploy source code to the integration/staging environment.
Apache Ant	Download from http://ant.apache.org/.
HP force-deploy-task (`force-deploy-task-bundle.zip`)	Download from the ALI Summary page. Unzip the zip file to `ant_install_dir/lib`).

Project deployment, testing, and report generation

Deployment of source code, testing, and the following report generation is ensured by the special Ant task *HP force-deploy-task*. In order to function, it is necessary to create the Ant build script “build.xml” (if it does not exist) in the root folder of a Force.com project similar to the following examples. See more details about attributes and elements in the table below.

- The following example deploys source code to a configured Force.com environment and runs all tests. Because all tests will be run, the report should contain code coverage of the whole project.

```
<project name="Sample usage of force-deploy-task" default="
deployAndTestAndReport " basedir=".">
        <target name="deployAndTestAndReport">
                <taskdef name="sfdeploy"
```

```
classname="com.claimvantage.force.ant.DeployWithXmlReportTask"/>
                <delete dir="test-report-xml" quiet="true"/>
                <sfdeploy
                username="username to force.com environment"
                password="password to force.com environment"
                serverurl="force.com server URL"
                deployRoot="path to source directory"
                runalltests="true"
                reportDir=" test-report-xml " />
        </target>
</project>
```

- The following example deploys source code to a configured Force.com environment and runs only tests that match the given pattern. In this case, Agile Manager will not be provided full code coverage.

```
<project name="Sample usage of force-deploy-task" default="
deployAndTestAndReport " basedir=".">
        <target name="deployAndTestAndReport">
                <taskdef name="sfdeploy"
classname="com.claimvantage.force.ant.DeployWithXmlReportTask"/>
                <delete dir="test-report-xml" quiet="true"/>
                <sfdeploy
                username="username to force.com environment"
                password="password to force.com environment"
                serverurl="force.com server URL"
                deployRoot="path to source directory"
                runalltests="false"
                reportDir=" test-report-xml ">
                <!-- Run only tests with file names that match this pattern -->
                <batchtest>
                        <fileset dir="src/classes">
                                <include name="*Test.cls"/>
                        </fileset>
                </batchtest>
        </target>
</project>
```

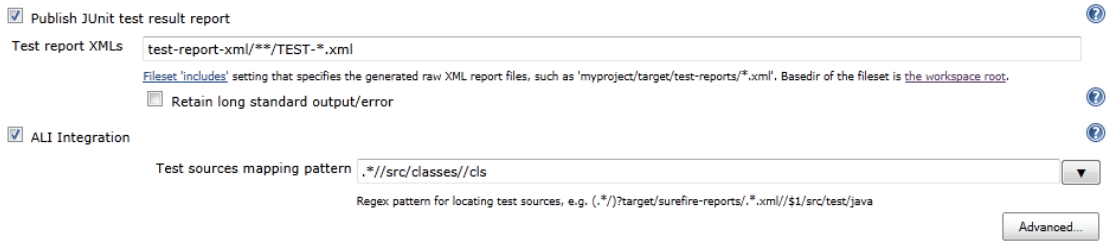
Description of the HP force-deploy-task (in the example defined as **sfdeploy**):

username	Attribute that defines the login name to the force.com environment
password	Attribute that defines the password to the force.com environment
serverurl	Attribute that defines the URL of the login page to the force.com environment

deployRoot	Attribute that defines the path to the source code directory that contains classes, triggers, and so on.
runalltests	Attribute that defines whether tests are started and project code coverage is reported. • **true**: All tests are started and code coverage is reported • **false**: Only tests specified by batchtest element are started. Code coverage is not provided.
reportDir	Attribute that defines where all reports will be stored.
batchtest	Element that specifies the tests that should be started. This works only if *runalltests*=**false**. *fileset* — Element that defines the file set of tests to run *dir* — Attribute that defines the directory where the tests are located *include* — Element that defines the classes of tests to run *name* — Attribute that defines the class name pattern of tests to run

Hudson/Jenkins Force.com configuration

1. Create a Free style job and configure the **SCM** and **Build Triggers** as needed.
2. Add the build step *Invoke Ant* and specify the targets which should be started (as in the example *deployAndTestAndReport*).
3. In the **Post-build Actions** section, configure as shown below.

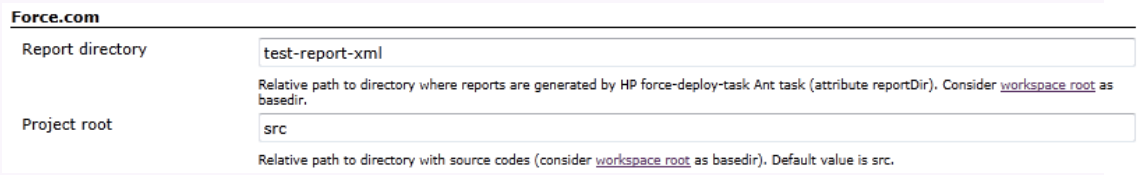


4. In the test report XMLs, replace the `test-report-xml` string with your actual report directory (attribute reportDir in force-deploy-task).
5. In the test sources mapping pattern, replace *src* with your actual path to the source directory.

The above configuration is sufficient for most cases. In other cases do one of the following:

If the **force-deploy-task** is defined in the *distributed* Ant script, which is called from the *main* Ant script	Configure the **Report** directory (the **reportDir** attribute value in the Ant script) in **ALI Integration/ Advanced**
If the source code directory, which contains folders, classes, triggers, and so on, is not in the **src** directory located directly in the workspace root	Configure the Project root

Configuration example:



ALI Extensibility

ALI provides an extensibility API that enables you to integrate Agile Manager with a custom build or SCM management system that is not supported out-of-the-box.

1. On the ALI Summary configuration page (**Workspace** > **ALI Summary**), under **Integration**, select **Custom Integration** and download the ALI SDK.
2. Develop Java connectors (called **providers** in ALI) to additional SCM and build management systems.

   Implementation classes, together with a simple descriptor containing basic metadata, such as name and version, are packaged in a zip file with a predefined layout.

Deploy your custom ALI plugin

1. Shut down Agile Manager. In a clustered system, make sure that you stop the Agile Manager service on all nodes.
2. Copy the plugin directory to the **%AGM_HOME%/repository/customerData/ali_plugins** directory. If this directory does not yet exist, create it.
3. Start Agile Manager.
4. Verify that the plugin is successfully deployed by browsing to the following URL: `http://agm-host:8080/agm/rest/ali/plugin-info?login-form-required=Y`

> **Note:** Depending on your Agile Manager deployment, you may need to modify the protocol

> and port number in this syntax.

If the deployment was unsuccessful, check the Agile Manager server log for additional details.

API integration

To integrate an application with Agile Manager, Site Administrators must register it on the Agile Manager **Integrations** > **API** configuration tab. For details, see "Register your application for integration" on page 227.

> **Note:** *Applications* in the context of API refer to the application you are developing, and want to integrate with Agile Manager.
>
> These are *not* Agile Manager applications, used within Agile Manager to organize a backlog.
>
> For details about Agile Manager applications, see "Work with applications" on page 82.

- Applications can be written in any language and on any platform that support REST. Request and return data are in JSON format unless specified otherwise.
- The API communicates via HTTPS.
- Your application server stores a client token used to authenticate on the Agile Manager API. This token, and the Client ID and Secret used to acquire it, should be stored with the same security considerations used to store passwords.

See "API authentication" on page 229 for an overview of authentication and the flow of requests.

General resources

Developers can retrieve and operate on the entities that are available in the Agile Manager graphical user interface. This includes releases, backlog items, workspaces, and more.

Unique resources

Developers can retrieve time sheet data stored in Agile Manager for specific users and date ranges. This is useful for project and portfolio management software, such as HP Project and Portfolio Management (PPM).

Only data entered to Agile Manager after installing Agile Manager 2.30 is available via API. Data entered in previous versions is not reflected in timesheet reports.

API Interactive Help

Use our interactive API help to explore supported entities, and to request data from Agile Manager. You can interact with Agile Manager directly from the Interactive Help, or by copying examples into your code.

Access the Interactive Help from the Agile Manager Help menu (**Help** > **API Interactive Help**), or from the **Integrations** > **API** configuration page.

1. To perform interactive operations, at the top of the page, enter your client ID and secret, and then click **Authenticate**.

   > **Note:** You can view available resources, operations, and parameters even without entering credentials. You must enter credentials only to submit operations.

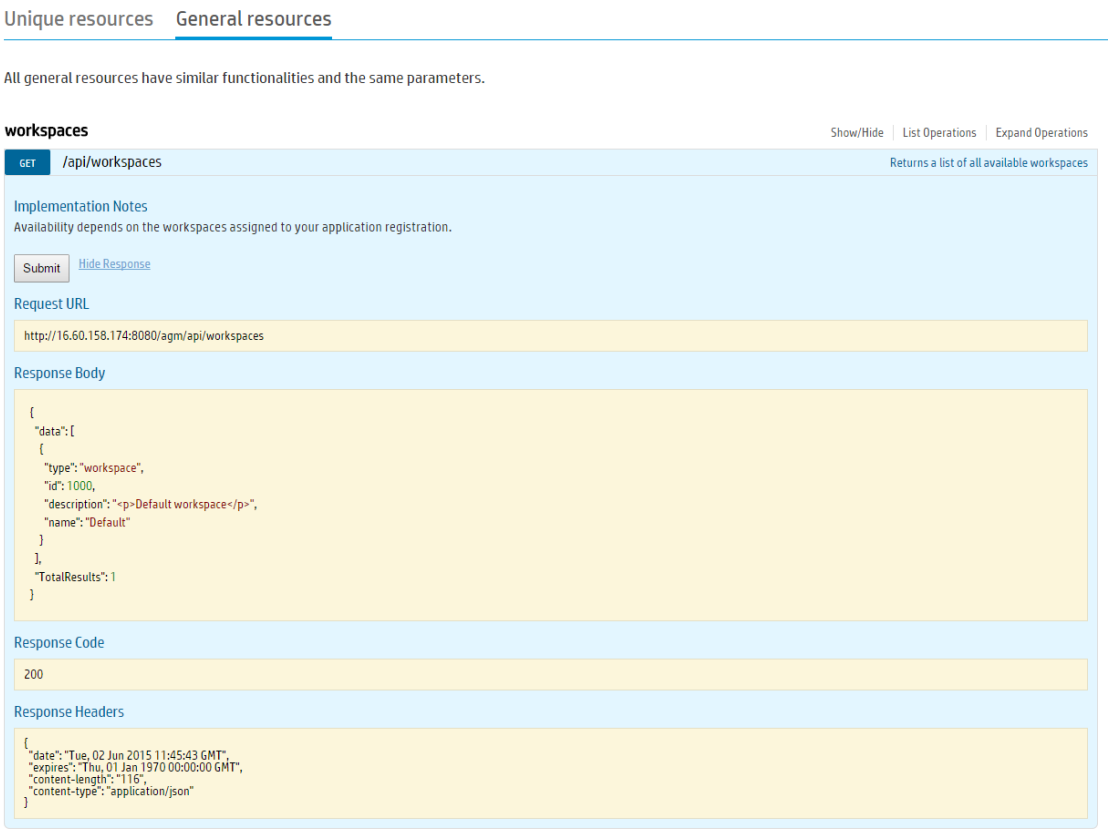
2. Expand unique and general resources and operations for more details about each one.

   The **timesheet** resource is a unique resource, with operations specified for generating timesheet data.

   Workspaces, **releases**, **backlog_items**, and more are general resources, all with the same operations.

3. For each operation, enter parameter values in **Value** column fields, as described in the **Description** column.
4. Click **Submit** to perform the operation using the parameters you've specified, or copy examples or syntax into your code.

View a sample response:



Caution: POST and PUT operations performed make *actual modifications in your data*, such as adding or editing backlog items or releases. You, and other users, can view these changes in Agile Manager.

For this reason, we've removed the **Submit** button for certain operations, such as **Delete**, to prevent you from making irreversible changes via the API Interactive Help.

See also:

"API exceptions" on page 236

For more details about integrating with HP PPM, see the Project and Portfolio Management documentation.

Register your application for integration

Click **Configuration** on the top right of the page, and select **Integrations** from the navigation menu on the left.

Tab: Integrations > API. Visible to **Site Administrators** only.

Register your application to integrate it with Agile Manager. Registration creates client credentials that you can use when authenticating your API over OAuth.

A **client** is similar to an Agile Manager user that accesses Agile Manager through the API. The client is assigned to specific workspaces and roles, just like other Agile Manager users. The client ID is generated automatically when you register a new application.

A **client secret** is generated for each client. This secret is like a user password, and you must record it securely. Agile Manager generates each secret once only, and the secret cannot be retrieved later. If needed, regenerate the secret to retrieve a new one.

What do you want to do?

Register a new application

1. Click **Add New**.
2. Populate the fields in the dialog box, and click **Register**.

   The workspaces and roles assigned to the client determine the items accessible by the client and API. For more details, see "User roles and permissions" on page 42.
3. From the **Client Secret** dialog box, copy the secret to a secure location for use in your API. You will not be able to access this secret from Agile Manager later on.

   The **Client ID** displayed in this dialog box is also displayed in the grid on the **Integrations > API** tab.

Edit an existing client

To modify a client's **Name**, **Description**, **Workspaces**, or **Roles**, click the Client ID link in the grid. You cannot modify the value of a client ID.

Regenerate a secret

1. Select the row for an existing client, and click **Regenerate Secret**.
2. Confirm that you do want to regenerate the secret. Doing so renders your current secret inactive, and you will need to update your secret in all relevant locations.
3. From the **Client Secret** dialog box, copy the secret to a secure location for use in your API. You will not be able to access this secret from Agile Manager later on.

Remove an existing client

Remove an existing client if it is no longer in use.

Select the row for an existing application, and click **Remove Credentials**.

Open the API Interactive Help (beta)

1. Click **API Interactive Help**.
2. In the top right corner of the Interactive Help page, enter your client ID and secret to access the API operations.

The Interactive Help is also available from the Help menu (**Help** > **API Interactive Help**).

For more details, see "API integration" on page 225.

API authentication

To support the API, Agile Manager implements a Client Credential flow in OAuth. For more information on the OAuth standard and the Client Credential flow, see The OAuth 2.0 Authorization Framework.

You need an OAuth access token in order to request data from Agile Manager. The access token must be included in the header of each API request.

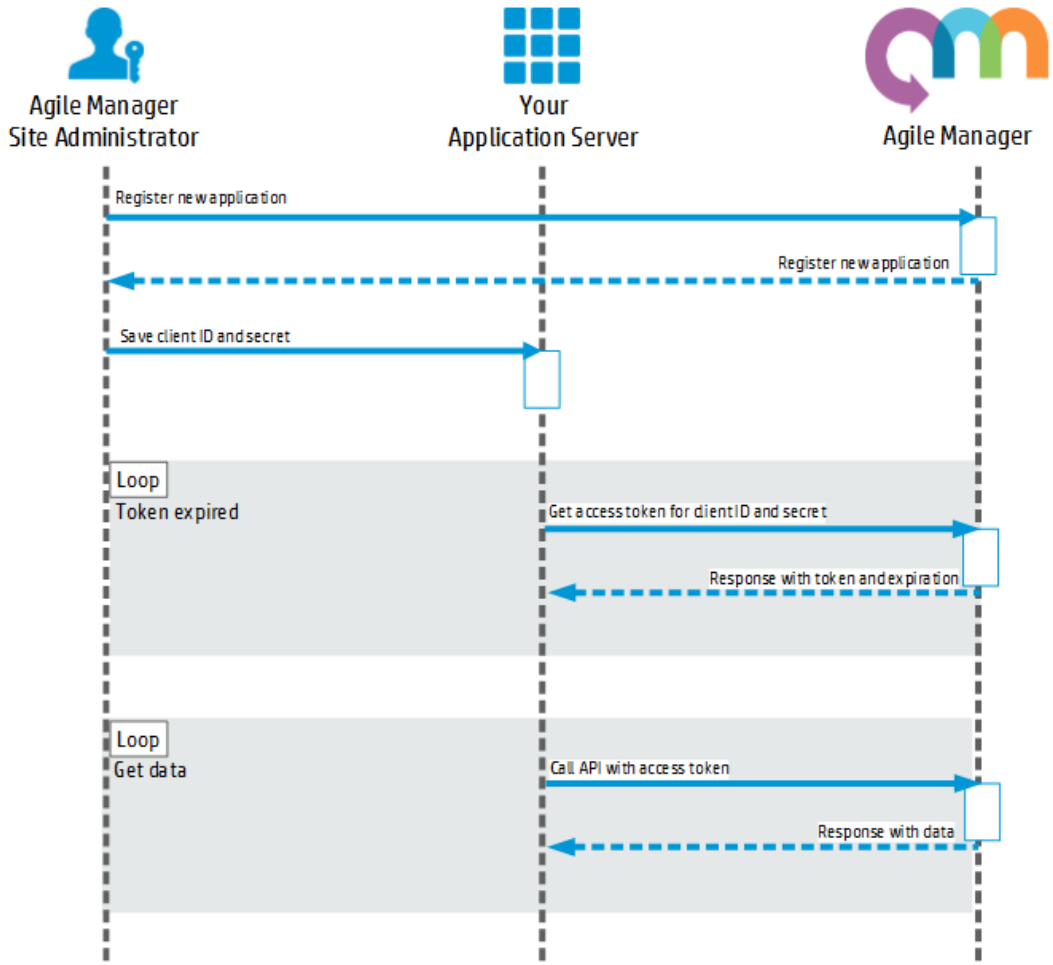
Note: If you use the API Interactive Help (beta), this token is obtained automatically for your when you log in to the Interactive Help using a pre-generated Client ID and Secret. For details, see "Register your application for integration" on page 227.

Get a token manually

1. Generate a Client ID and Secret on the **Integrations** > **API** configuration page. For details, see "Register your application for integration" on page 227.
2. Send the Client ID and Secret to the token endpoint resource, Access Token.
3. When the current token expires, send the Client ID and Secret to the Access Token resource again to get a new token.

Caution: The access token and the Client ID and Secret used to acquire it should be stored with the same security considerations used to store passwords.

Click to view a diagram of the Agile Manager API authentication sequence.



Access token resource

The token resource returns data containing the access token required in the header for other requests in the API.

Access token URI: `/agm/oauth/token`

This resource is the **Authorization Endpoint** described in the The OAuth 2.0 Authorization Framework, and can be returned by either a GET operation or POST operation.

Return values

The JSON array for each record contains the following fields:

access_token	The token to be passed as the value of the Authorization header in requests to resources in /agm/api/
token_type	Generally "bearer"
expires_in	Generally about one hour (seconds)
scope	Generally "read trust write"

Return Value Example

```
{
	"access_token":"777856517_09a705b4-a4b7-4812-98b6-136210de7e4c",
	"token_type":"bearer",
	"expires_in":3599,
	"scope":"read trust write",
}
```

Example getting token with GET Operation

```
GET http://my-server:8080/agm/oauth/token?grant_type=client_credentials HTTP/1.1

Accept: application/json

Authorization: Basic‹base 64 representation of OAuth Client ID:OAuth secret›
```

Example getting token with POST Operation

```
POST http://my-server:8080/agm/oauth/token HTTP/1.1

Accept: application/json

Content-Type:application/x-www-form-urlencoded

client_id=‹client ID› & client_secret=‹secret› & grant_type=client_credentials
```

GET query specifications

GET queries are used to retrieve a specific set of items from Agile Manager.

This topic provides specifications for "Queries" below, "Pagination" on page 234, "Fields" on page 235, and "Ordering (Sorting)" on page 235. Within in each syntax description below, click hyperlinks links to drill down for further details.

The following is an example of a full query within a GET request.

```
http(s)://…/agm/api/workspaces/1000/releases?query=”id>1001”&order-by=end_
date&fields=id,name,end_date&offset=1&limit=3
```

This example returns releases with IDs greater than **1001** from the default workspace (workspace ID = 1000).

The releases returned are sorted in ascending order, by the release end date.

The fields included in the data are limited to the release **ID**, **name**, and **end date**.

Only three releases are returned, starting from the second release in the workspace.

Queries

Each query consists of a **query statement**.

- Query statements are surrounded in double-quotes ("), and consist of one or more query phrases and any relevant logical operators.
- You must have at least one query phrase in each statement. If you add additional query phrases, you must add a logical operator before each one.

Use the following query syntax:

```
query="<query phrase>[[<logical operator><query phrase>]]"
```

The following example shows a query statement that includes multiple query phrases.

```
http://.../backlog_items?query="id>100;status='open';(rank>10||rank<20)"
```

This example returns all backlog items that have an ID greater than **100**, who's status is **Open**, and are ranked between **10** and **20**.

Query phrases

You can nest query phrases within statements as needed, as well as promote specific phrases using parentheses.

Use an exclamation mark (**!**) as an optional **negate keyword**.

Use the following syntax for each query phrase:

```
[negate keyword]<field name><comparison operator><value>
```

The following example shows a query phrase which would return all items that have an ID other than 100 or 101.

```
"!(id=100 || id=101)"
```

Comparison operators

Comparison operators are used to separate between field names and their values, and include the following:

Operator	Description	Examples
=	Equals to	`id=1001` returns all items with an ID of **1001**
<	Less than	`id<1001` returns all items with an ID less than **1001**
>	Greater than	`id>1001` returns all items with an ID greater than **1001**
<=	Less than or equal to	`id<=1001` returns all items with an ID less than or equal to **1001**
>=	Greater than or equal to	`id>=1001` returns all items with an ID greater than or equal to **1001**

Values

Values are either numerical or string based.

String values must be surrounded by single quote (`'string'`).

Any single quotes or double-quotes within a string value must be preceded by a backslash (\).

The following example shows a query that will return all backlog items named As a user I want to access the "configuration" area

```
"name='As a user I want to access the  \"configuration\" area'"
```

Logical operators

Logical operators are used to separate between query phrases or query statements, and include:

Operator	Description	Example
;	And	"id>100;status='open'" returns all items with an ID greater than **100** and a status of **open**
||	Or	"!(id=100 || id=101)" returns all items that have an ID other than **100** or **101**.

Pagination

Pagination allows you to retrieve a limited collection of results from Agile Manager, as well as an offset of results.

> **Note:** In the response, the **Data** section includes only items defined by the offset and limit values.
>
> However, the **TotalResults** value reflects the entire number of items returned from the query, without the offset and limit.

The following example returns 10 domains, at places 40-49:

```
*** Request ***
GET /backlog_items?offset=40&limit=10
Accept:             application/json
Host:               agm.com:8080

*** Response ***
HTTP/1.1 200 OK
Content-Encoding:   gzip
Content-Type:       application/json;q=0.9
Date:               Mon, 27 Mar 2014 12:11:05 GMT
Server:             Jetty(7.5.4.v20111024)
{
       "data": [
              ...
       ],
        "TotalResults":40
}
```

Pagination keywords include:

Limit

Limits the collection of results to a specific number of items.

If no limit parameter is provided, the results are limited to default maximum limit, which is 2000.

Offset

Defines the starting point for the collected results.

If no offset parameter is provided, the results start from **0** (zero).

If you use an offset value, you must also use a limit value. Any offsets provided without a limit value are ignored.

Fields

Query fields allow you to limit the set of fields returned with the results, using the following syntax:

```
fields=<field name>[[,<field name>]]
```

Field names must be separated by commas (**,**)

Some fields are always returned, including: **type**, **subtype**, and **id**

If no fields are defined, all of the fields for the specified resource are returned.

Tip: To view the full list of fields for an entity submit a GET operation for that entity.

The following example shows a set of fields.

The data returned with this query will include details about the rank, name, and end_date fields only, in addition to the fields which are always returned.

```
fields=rank,name,end_date
```

Ordering (Sorting)

Allows you to sort a collection of results retrieved, using the following syntax:

```
order-by=[<direction>]<field name>[[,[<direction>]<field name>]]
```

By default, results are sorted in ascending order. This default is used whenever the direction is left empty.

Use a minus sign (**-**) to sort results in descending order.

The following example shows a query that will return items sorted:

- First by *name*, in ascending alphabetical order;
- And then sub-sorted by *status*, in descending order.

```
http://.../backlog_items?order-by=name,-status
```

Back to top

API exceptions

Exceptions are returned in format `application/json`. The `Id` element refers to the exception type. The `Title` provides more specific information.

Your application handles REST exceptions by extracting the exception `Id` and `Title`, and taking application-appropriate action.

Example

```
{
	"Id": "qccore.required-field-missing",
	"Title": "Missing required field: severity for entity defect",
	"ExceptionProperties": [{
		"Name": "field-name",
		"StringValue": "severity"
	}, {
		"Name": "entity-name",
		"StringValue": "defect"
	}],
		"StackTrace": null
}
```

Glossary

A

Acceptance test
A test defined by the developer or customer to verify that the application as delivered meets the conditions of a user story.

Action item
In Sprint Closure, a specific activity that is called for at the end of a sprint. Action items can be generated automatically from the sprint retrospective, and can be converted in turn to user stories that are added to the backlog.

Actual effort
The total number of hours invested and remaining for all tasks in a backlog item.

ALI Dev Bridge
The ALI Dev Bridge is a simple mini-web application that connects HP Agile Manager hosted in a SaaS environment to Source Code Management systems and Build Management systems hosted in your local development environment.

Application
One of the components being developed in the framework of your project. An application is defined by the features you assign to it.

Application (API)
An application or program you develop, and integrate with Agile Manager over the Agile Manager API. Register your application in the configuration area to create client credentials used when authenticating your API over OAuth.

Archive
Backlog items removed from the product backlog. Adminstrators can archive themes and features, or remove items from the archive. Archiving an item also archives any related child items.

B

Backlog
A list of work items planned to be handled during a specific stage of the product development life cycle. Can refer to the product, release, or sprint backlog.

Backlog item
A work item that is listed in the product, release, or sprint backlog. A backlog item can be either a user story or defect.

Build
A process to produce the key deliverables of software development.

Build Agent
A Build Agent is a set of scripts or proprietary applications installed on a Build Server configured for listening on a Build System. When appropriate, changes on the Build Server are pushed to ALI.

Build configuration
A build configuration specifies how and with what tools the build is to be built.

Build server
A computer used to create a build. The server automatically runs the unit tests periodically or even after every commit and reports the results to the developers.

C

Client ID
An ID created for a client application that you wish to integrate with Agile Manager via the Agile Manager API. Clients are assigned to specific workspaces and roles, similar to Agile Manager users.

Client secret
A password for a specific client, created for an application you wish to integrate with Agile Manager. Record your client secret securely, because it is like a password, and once generated, cannot be retrieved again. If you need to, generate a new secret to replace your current secret.

Closed defects (ALI)
ALI metric that measures the number of defects closed in a build.

Code change
Any change made to the code that was committed to the SCM repository. ALI metrics count the number of lines of code changed in a build. Code changes are often grouped by their association with user stories, defects, or neither.

Code coverage
ALI metric that measures the percentage of code in a build that is covered by unit tests.

Committer
A developer who committed code changes.

Custom field
Fields defined by administrators in the configuration area. Custom fields are created at the site and workspace level, and can apply to user stories, defects, all backlog items (user stories and defects), themes, or features.

Cycle
A sub-division of ALM releases. In ALM, requirements and defects can be assigned to cycles within a release. When synchronizing releases, ALM cycles are matched with Agile Manager sprints.

Cycle redundancy checks (Synchronizer)
Verify whether the updates detected in synchronized records were made in fields specifically mapped for the selected link. If the changes were made in fields that are not mapped, the record is not synchronized.

D

Defect
A fault or bug detected in the application under development. Defects are included in the product backlog as backlog items along with user stories. Defects can be linked to user stories, and can be associated with features.

Detected defects
ALI metric that measures the number of defects detected in a build.

Dominant side / Dominant endpoint
When synchronizing data between Agile Manager and ALM: The endpoint whose data is used in cases of conflict, where the same entities were modified both on Agile Manager and on ALM.

E

Endpoint
An instance of ALM or Agile Manager connected that is synchronized with another endpoint using HP ALM Synchronizer. ALM endpoints can only connect to Agile Manager endpoints, and Agile Manager endpoints can only connect to ALM endpoints.

Entity (Synchronizer)
The type of data synchronzied in a Synchronizer link. Entity types include releases, requirements/user stories, and defects.

Entity link
A link between two entities in Agile Manager or ALM, such as a link between a requirement and a defect.

F

Feature
A functional area in an application. Several features can be grouped together under a theme. A feature is fulfilled by the user stories associated with it.

Feature team
Teams created and used only in the scope of a single release.

Full synchronization
A full comparison and update of the records in each endpoint, including deleted records.

G

Group story
After you break a user story it is converted to a group story. The group story contains the parts that you define when breaking the original story. To view group stories, on the Product Backlog > Backlog page, select the Group Stories View.

I

Identity mappings (Synchronizer)
Unique IDs that are stored in a mapping table for records in each endpoint. This mapping table records the correspondence between each pair of records.

Incremental synchronization
Synchronizes data between two endpoints for records that were created or modified since the last synchronization task.

Integration Bridge
The platform that enables two-way communication between Agile Manager and on-premise applications located behind firewalls, such as HP ALM.

Integration Bridge service
A Windows service or linux daemon that makes sure the Integration Bridge is automatically started when you system starts.

K

KLOC
Thousand Lines Of Code. Used to measure the amount of effort used in the item being measured.

L

Link destination endpoint
The Synchronizer link endpoint to which data is synchronized. Data in the destination endpoint is updated according to the data in the source endpoint and the mapping's settings.

Link source endpoint
The Synchronizer link endpoint from which data is synchronized. Data in the source endpoint for the mapping remains unchanged

P

Planned effort
The total number of hours estimated for all tasks in a backlog item.

R

Release
A group of changes to the application that will be available for distribution at the same time. You can assign user stories and defects to the release.

Role
Each user in the project is assigned a role. The role defines the user's read/write permissions in different areas of Agile Manager.

S

SCM
Source Code Management is the management of changes to documents, computer programs, large web sites, and other collections of information.

SCM agent
An SCM agent is a set of scripts or proprietary applications installed on a SCM server configured for listening on a SCM system. When appropriate, changes on the SCM server are pushed to AGM.

SCM branch
A set of files under version control may be branched or forked at a point in time so that, from that time forward, two copies of those files may develop at different speeds or in different ways independently of each other.

SCM repository
A database used to enable the collaborative development of large projects by multiple engineers.

Sprint
A time period, measured in days or weeks, in which teams work on a set of backlog items planned to that sprint and team.

Sprint capacity

1. The number of hours a team member is available to work in a sprint. The sprint capacity is based on the team member's daily work hours multiplied by the number of his/her work days in the sprint. 2. The sum of all team velocities, in story points, of teams participating in a sprint.

Story points

A method for estimating the size of a user story or defect. Story points compare one story to another to determine a relative size and then assign points denoting that size. A team's estimated sprint velocity is then used to estimate how many story points the team can deliver in a sprint.

Synchronization link

Connects an ALM endpoint and an Agile Manager endpoint and synchronizes data between them.

T

Task

One of the work items that should be carried out in order to implement a user story or defect. A user story or defect is considered done when all its tasks are completed.

Team

A group of people that work together on a sprint. Teams are defined at the release level. A team can include members from different departments (such as development, QA and technical writers) who work together on the same backlog items. The amount of work a team can deliver in a sprint is estimated in story points.

Team member

A user that is assigned to a team. A user can be a member on a number teams simultaneously. This can represent a shared resource that contributes to several teams, such as a technical writer.

Test success

ALI metric that measures the success rate of unit tests run on a build.

Theme

A top-level objective or a high-level functional area that is fulfilled by the features associated with it. A theme may span several applications.

Time sheet data (API)

Data available for retrieval from Agile Manager through the API. Time sheet data includes changes made to the Invested (hours) field for each assigned task, for specified users, and within a specific time frame.

U

User story

A basic action that a user should be able to carry out in the application. User stories are added to the product backlog and estimated in story points. User stories are typically written using the following format: 'As a <role>, I want <goal/desire>'.

V

Velocity
The number of story points a team is estimated to complete or has completed in a sprint. The estimated team velocity is used as a basis to calculate the total sprint or release capacity.

W

Widget
A graph or mini application that can be displayed in a dashboard. Widgets are available from the Widget Gallery.

Worked on defects
ALI metric that measures the number of defects worked on in a build.

Workspace
A top-level layer of data visiblity that enables you to apply scaled agile methodology in an enterprise organization. Users must be assigned to specific workspaces, and can view and access only data within the assigned workspaces.

Workspace team
Teams created at the workspace level, and reused in multiple releases.

Send Us Feedback

Let us know how we can improve your experience with the HP Agile Manager User Guide.

Send your email to: SW-doc@hp.com


Java™
COMPATIBLE