# Project Management v3

## Using ServiceNow's Improved Project Management Application

PDF generated using the open source mwlib toolkit. See http://code.pediapress.com/ for more information.
PDF generated at: Wed, 29 Nov 2017 20:50:41 PST

# Introduction to Project Management

# Project Management

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Project Management* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**'

## Overview

The Project Management application is a suite of tools that aids in managing projects, tasks, and resources. It provides the ability to create and manage projects of all sizes, from small projects with a few tasks to large portfolios of projects that contain complex tasks with various relationships and dependencies.

Project management includes tools to help you create, view, and manage projects:

- Project workbench gives project managers the ability to manage the different aspects of a project from a single page. This workbench supports both Project Management and Application Lifecycle Management applications, allowing for a hybrid approach to project management. Project managers can create projects that combine both Waterfall and Agile methodologies by using Waterfall, Agile, and Test phases. The project workbench is available starting with the Fuji release.
- Project templates define the basic structure of a project, including project tasks and sub-tasks, attachments, and other project information. The project template feature gives project managers a simple way to create, save, and reuse this project structure.

Project Management also includes features that enable you to achieve your project goals in alignment with the other activities your organization is managing with ServiceNow, such as:

- Integration with other features and applications on the ServiceNow platform, such as change management, resource management, and reports.
- Easy-to-read Gantt charts and Work Breakdown Structure lists that help you visualize large projects with complex relationships and dependencies.

## Video Tutorial

This video provides a brief tour of the Project application in the Eureka release.

## Key Terms

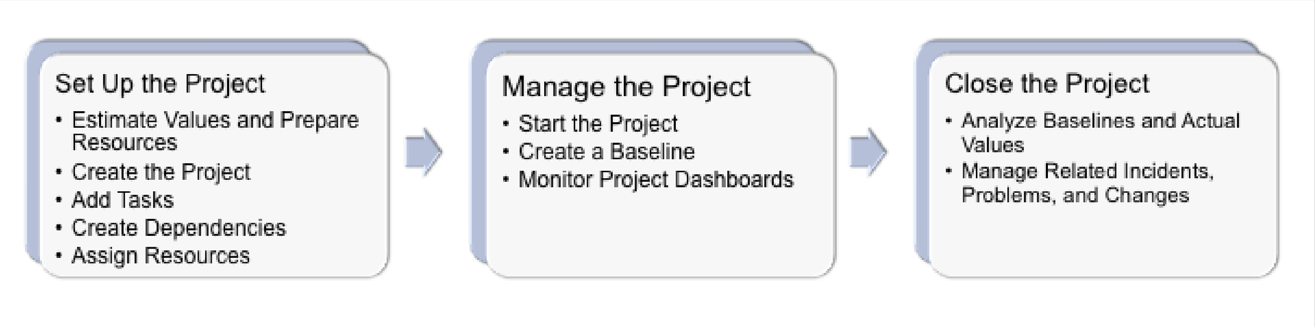
- **Portfolio:** a collection of projects managed as a group to achieve strategic and operational objectives.
- **Project:** any planned, collaborative effort that is designed to achieve an objective.
- **Agile project:** any planned, collaborative effort that is designed to achieve an objective and uses Agile.
- **Project workbench:** a single page that presents project information in two panes. The the upper pane includes a project timeline and displays the project phases and milestones. The lower pane presents details about the currently selected phase in either a list view or visual task board.

- **Phase:** one stage or one segment of a project. Three types of phases can be added to the timeline in the project workbench:
  - **Waterfall phase:** a waterfall phase contains project tasks. A project can have multiple waterfall phases.
  - **Agile phase:** an agile phase contains stories. A project can have one Agile phase.
  - **Test phase:** a test phase contains test cases and can also include a team assignment. A project can have one test phase.
- **Story:** a brief statement of a product requirement or a customer business case that is used in the scrum method of agile software development. Typically, stories are expressed in plain language to help the reader understand what the software should accomplish.
- **Task:** a unit of work within a project. Projects typically contain several tasks.
- **Test case:** a collection of related tests. A test case is saved as part of a test suite and can be added to a test plan.

# Basics of Project Management

The Project application helps you plan and track projects, plus it integrates with other ServiceNow applications. For example, if an incident, problem, or change is large enough to require an entire project to manage, create projects from an incident, problem, or change form.

There are several paths available to manage a project. The best path usually depends on business needs. The steps below are designed to get a project up and running with the minimum amount of effort. Alternative methods to these procedures are also explained.


Set Up the Project
- Estimate Values and Prepare Resources
- Create the Project
- Add Tasks
- Create Dependencies
- Assign Resources

Manage the Project
- Start the Project
- Create a Baseline
- Monitor Project Dashboards

Close the Project
- Analyze Baselines and Actual Values
- Manage Related Incidents, Problems, and Changes

## Set Up the Project

Setting up a project involves deciding on an approach for creating and linking project tasks and making sure the necessary users and groups are created in ServiceNow so you can assign them to project tasks.

### Plan the Project

Before creating a project, consider the following questions and issues:

- **Do you want a top-down or bottom-up approach to tasking?**

  Top-down tasking involves creating a project first, then identifying major project phases. Later on, phases can be broken down into tasks and subtasks. The emphasis is on creating estimates for high-level items such as phases and parent tasks and then building the project down from there toward a more detailed level. Use caution when creating tasks for top-down tasking. If you first create a project and then create a task under it with a start-on date later than the project's start date, the project shifts later to start on the task start date. The Project application supports bottom-up tasking better.

  Bottom-up tasking involves creating several sets of small tasks and estimating task items such as effort, cost, and duration. These estimations are then aggregated into high-level parent tasks (rollup tasks) and phases. The emphasis is on estimating smaller chunks of work as accurately as possible first, then letting those estimations roll up into parent tasks, phases, and the project itself.

- **Is the project part of a larger portfolio of projects?**

  Also consider portfolio planning and how the project relates to similar projects or initiatives.

- **What types of dependencies will the tasks have with other tasks?**

  The Project application supports only finish-to-start dependencies.

- **Can milestones and project baselines help manage a project?**

  A milestone is a project task with a duration of 0. Use milestones to indicate important dates in a project. If necessary, create dependencies between tasks and milestones so that a task does not start until a milestone has been reached.

  A baseline is a snapshot of each task's current planned start and end dates at the time the baseline was created. An line appears under each task on the Gantt chart for the original planned start and end dates. The line appears shifted to the left or right depending on whether the task was started early or late. If tasks slip to later dates, the baseline indicator provides an easy way to see how severe the delays will be.

- **Have the necessary skills, groups, and resources been created in ServiceNow?**

  If project tasks will be assigned to different groups or individual resources with the required skills, create users and groups and configure the Skills Management application.

- **Does an existing incident, problem, or change justify creating a project in order to track it?**

  Of these record types, a change is most likely to lead to activities that should be tracked as a project.

- **Do you want to track project costs?**

  Estimate group resource costs before starting the project and then track the actual cost of each user resource from time cards. The Project application can also calculate the costs of affected CIs in a project. The Project Management Costing add-on is required to track costs.

- **What goals do you want the project to achieve?**

  Every project should have at least one goal. Project goals are saved in the Goal table and can link to any task. In a typical scenario, link one goal to each project and keep the goal's **State** field up to date.

## Create the Project

After choosing an approach and gathering initial estimates for the planned start date, estimated cost, and a well-defined business case, create the project in the Project application or in the project workbench.

## Add Project Tasks, Dependencies, and Relationships

After creating a project record, create tasks.

- For top-down planning, create a task that you already know will include several child tasks. Then create the child tasks and specify that they are child tasks of the first task you created.
- For bottom-up planning, create tasks for the smallest units of work first. Then you can create *intermediary* parent tasks that cover a group of related child tasks. For example, if there are five sequential tasks that comprise a phase of a project called *install database*, create the five tasks first. Then create another task called *Database installation* and make it the parent task of the five tasks. Rollup calculations, such as **Planned duration**, for the *Database installation* task are automatically calculated based on the child tasks.

It is easiest to build task relationships and dependencies while creating sets of tasks.

- A dependency means one task is forced to start after another task finishes. This is the only type of dependency ServiceNow supports.

- A relationship means a parent-child relationship whereby several subtasks are configured under a parent task or phase, which rolls up fields like **Planned duration** and **Estimated cost**.

Use the Gantt chart in conjunction with task forms and related lists to build relationships. Add milestones based on the project's major events and create dependencies between milestones and tasks, if necessary. See Project Task Relationships and Dependencies and Gantt Chart for more information.

Also set up notifications to alert project task assignees when their tasks move to the **Work in Progress** state. See Creating Project Tasks for more information on creating tasks.

### Assign Resources or Assignment Groups to the Tasks

User resources are the individuals in an organization who are assigned to project tasks. You can manage your resources with resource plans in the Resource Management application, starting with the Dublin release. In versions prior to the Dublin release, or if you are not using the Resource Management application, you can select resources from users or groups.

See Working with Resource Plans if you are using the Resource Management application. Otherwise, see Task Resources.

### Add the Project to a Portfolio

A portfolio is a group of related projects. If the project is related to other projects, create a portfolio and add the project. The Project application provides a useful portfolio view that makes its easy to report on the status of all projects in a portfolio. Portfolios also include demands starting with the Fuji release.

## Manage the Project

After the preceding steps are complete, the project can be started. To measure the project against initial estimates, create a baseline, which is a snapshot of the entire project including all planned dates for all project tasks and milestones. The project manager can manage a project from the project workbench starting with the Fuji release.

### Start the Project

Start the project by clicking **Start project** on the Project form or changing the project state to **Work in Progress**. Starting the project changes the **State** field on the Project form to **Work in Progress** and changes the **Actual start date** of the project to the current date. See Starting a Project for more information.

### Monitor the Project and Customize Dashboards

ServiceNow provides the ability to update important project status information, such as the number of milestones slipped. It also provides summaries for cost, scope, project risk, and so on. Modify this information as needed with the **Portfolio View** related list on the Portfolio form and display this information on the **Project Overview** homepage. In addition, use the project reports installed with the application, such as **Active projects** or **Projects (by priority)**, to show important project information.

When the project is underway, continue to access project records and make changes to several items, including costs, priority, schedule, and planned values that are not rollups. Keep detailed project records for risks and issues and refer back to them after a project is complete. Also create baselines along the way to easily see if any project phases or tasks are slipping at the time you create the baseline.

- See Project Reporting for more information on the available reporting options.
- See Project Portfolio Management for more information on viewing a summary of all project information in a portfolio, including completed and slipped milestones.
- See Updating a Project in Progress for more information on what is necessary while a project is underway.

## Close the Project

When the project is complete, change its state to **Closed complete** on the project form. When a project is in the closed state, the Project application calculates actual values like **Actual duration**.

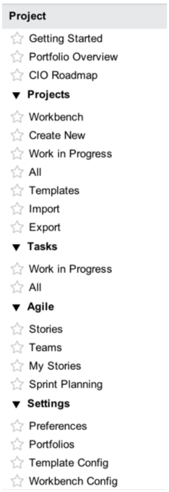
Post-project activities include analyzing project baselines and actual values and generating a final project dashboard. If the project was successful and can be used as a template for future projects, make a copy of it.

If the project was created from a change, incident, or problem record, there are several other activities you may need to perform in ServiceNow. See Closing a Project for more information.

# Menus and Modules

Activating this feature adds the Project Management menu to the application navigator with the following modules.

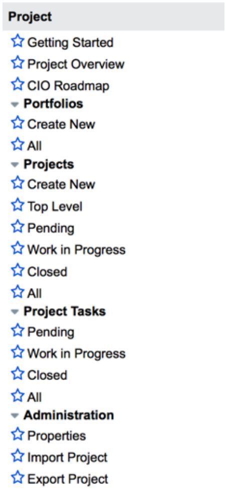
## Fuji Menu



The Project application (Fuji)

- **Getting Started:** Access the wiki documentation for the Project application.
- **Portfolio Overview:** Open the Project Overview homepage, which contains several built-in reports.
- **CIO Roadmap:** Open the CIO Roadmap.
- **Projects**
  - **Workbench:** Open the project workbench.
  - **Create New:** Create a new project.
  - **Work in Progress:** View projects currently in progress.
  - **All:** View all project records.
  - **Templates:** View project templates.
  - **Import:** Import a Microsoft Project.
  - **Export:** Export a ServiceNow project to be used in Microsoft Project.
- **Tasks**
  - **Work in Progress:** View project tasks currently in progress.
  - **All:** View all project task records.
- **Agile:** this module appears if the Project application is activated as part of the Project Portfolio Suite.
  - **Stories:** View all stories.
  - **Teams:** Display the Teams list, which shows a list of current teams.
  - **My Stories:** View a list of stories assigned to the current user.
  - **Sprint Planning:** Open the Sprint Planning page for the selected team.
- **Settings**
  - **Preferences:** Edit settings for project management properties.
  - **Portfolios:** View a list of portfolios.
  - **Template Config:** Open the project template configuration page.
  - **Workbench Config:** Open the workbench configurations page.

## Eureka and Prior Versions



The Project application (Eureka)

- **Getting started:** Access the wiki documentation for the Project application.
- **Project Overview:** Open the the Project Overview homepage, which contains several built-in reports.
- **Portfolios**
  - **Create New:** Create a new portfolio.
  - **All:** View all portfolio records.
- **Projects**
  - **Create New:** Create a new project.
  - **Top Level:** View all projects that do not have a parent project.
  - **Pending:** View pending projects.
  - **Work in Progress:** View projects currently in progress.
  - **Closed:** View projects already closed.
  - **All:** View all project records.
- **Project Tasks**
  - **Pending:** View pending project tasks.
  - **Work in Progress:** View project tasks currently in progress.
  - **Closed:** View project tasks already closed.
  - **All:** View all project task records.
- **Administration**
  - **Properties:** Edit settings for project management properties.
  - **Import Project:** Importing Projects from Microsoft Project.
  - **Export Project:** Exporting a ServiceNow project to be used in Microsoft Project.

# Integration with Project Portfolio Suite

Project Management can be used as a separate application or it can be activated as part of the Project Portfolio Suite (PPS). This application provides a simplified, team-oriented approach to IT development by combining several individual applications and integrating the different components of the project development lifecycle.

# Activating Project Management

Administrators can activate the Project Management plugin.

Upgrading to the Dublin release does not automatically upgrade you to the v3 application. If Project Management v2 is active and you want to upgrade to v3, read the upgrade instructions.

**Click the plus to expand instructions for activating a plugin.**

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

The Project application can also be activated as part of the Project Portfolio Suite.

# Enhancements

## Fuji

- The Project Management application is integrated with Project Portfolio Suite (PPS).
- Components of the SDLC (Scrum Process) are also integrated with Project Management to enable a project management approach that combines the Waterfall and Scrum methodologies.
- The project workbench provides a central location for managing projects and project phases. The workbench supports both the Project Management and Application Lifecycle Management applications, allowing for a hybrid approach to project management.
- The project calculation engine supports manual project calculation in addition to auto calculation.
- The composite field combines information from two different fields, typically a project or project task number and a short description.
- Project templates define the basic structure of a project and enable the project manager to create, save, and reuse project structure. This feature is available with the Fuji release.
- The IT Finance application adds a Finance view to the Project and Portfolio forms. The Finance view adds a chart that shows expenses that were allocated to the project or portfolio, shown by the financial bucket that the expense is associated with. See IT Finance for more information.

## Eureka

- Project managers can export ServiceNow projects to Microsoft Project, where the project can be managed and then imported back into ServiceNow.
- New business rules populate project-specific fields on non-project tasks with default project task data when these tasks are added to a project.
- The Project application automatically creates a new record in the Portfolio Project table for all new projects. This allows project managers to add a project to a portfolio by associating the newly created record with the portfolio.

## Dublin

- A new version of the Project application is available: version 3. See Project Management v2 to v3 Upgrade for upgrade information and instructions.
- Resource planning can be accomplished through the Resource Management application.
- The core project engine has been improved for the new version of the application. This results in better performance, usability, and scalability, especially with large projects.
- The Project application includes a default schedule that is applied to all new projects and project tasks. The schedule uses a 40-hour work week, from 8 A.M. to 5 P.M. with an hour break at noon.
- Project managers can now link existing change request records or create new change request records to link to project tasks. This feature links change management with project management.
- The Project application framework that supports importing from Microsoft Project 2010 was improved.
- State changes now roll down from the project to project tasks and from parent tasks to child tasks. For more information, see Project Task Relationships and Dependencies.
- Project managers can no longer modify most of the form fields on parent tasks. This enforces the concept that all parent tasks should derive aggregate values from their child tasks.
- A work breakdown structure (WBS) is available for project managers, and a new view, **WBS**, is available on the Project form.
- For v3, the functionality of the project management costing add-on has been moved to the Cost Management plugin.

- Projects can be included in multiple portfolios.

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_ProjectApplicationOverview.html

# Activating and Upgrading

# Installed with Project Management

## Overview

The following components are installed when you activate the project management v2 or v3 plugin. The v3 plugin is available starting with the Dublin release. Starting with the Eureka release, only the v3 plugin is available.

- Tables
- Plugins
- Script Includes
- Client Scripts
- UI Policies
- UI Actions
- Business Rules
- Schedule Pages
- Schedules
- Properties
- User Roles

Demo data is available with project management and provides sample projects, portfolios, and tasks.

## Tables

The project management v2 and v3 plugins add or modify the following tables:

| Table | Table Content |
|---|---|
| Planned task [planned_task] | Adds the following:<br>• **start_date:** Planned start date.<br>• **end_date:** Planned end date.<br>• **duration:** Planned task duration.<br>• **work_duration:** Actual task duration.<br>• **percent_complete:** Completion percentage of the task, either manual or calculated.<br>• **time_constraint:** Time constraint for the task, either **Start ASAP** or **Start on a specific date**. |
| Task Relationship [task_rel_task] | Predecessor and successor tasks in a dependent relationship. |
| Planned task Relationship [planned_task_rel_planned_task] | Predecessor and successor tasks in a dependent relationship, including task lag values. |
| Project [pm_project] | All projects. |
| Project Task [pm_project_task] | All tasks used in projects. |
| Baseline [planned_task_baseline] | All baselines. |
| Baseline Item [planned_task_baseline_item] | All tasks that are part of baselines. |
| Portfolio [pm_portfolio] | All portfolios. |
| Portfolio Project [pm_portfolio_project] | All projects in the **Portfolio View** related list of a portfolio. These projects are referred to as portfolio projects and are used for reporting purposes. |

| | |
|---|---|
| Portfolio Project Goal [pm_portfolio_goal] | All goals for a portfolio project. |
| Pm Portfolio Group Resource [pm_portfolio_group_resource] | All group resources for a portfolio project. |
| Portfolio Project Issue [pm_portfolio_issue] | All issues for a portfolio project. |
| Portfolio Project Risk [pm_portfolio_risk] | All risks for a portfolio project. |
| Portfolio Project Relationships [pm_m2m_portfolio_project] | All relationships between a portfolio and each project in the portfolio. |
| Project Task Link [pm_project_task_link] | All records for linked changes (installed with the v3 plugin available starting with the Dublin release). |
| Project Template [project_template] | All project templates (available with the Fuji release). |
| Project Template Configuration [project_template_config] | All project template configuration items. |
| Project Template Task [project_template_task] | All project template tasks (available with the Fuji release). |
| Domain [sys_domain] | All domains. |
| Issue [issue] | All issues in the **Issues** related list for a project. |
| Goal [goal] | All project goals. |
| Risk [risk] | All risks in the **Risks** related list for a project. |

# Plugins

The following plugins are activated along with the project management v2 and v3 plugins if they are not already active.

| Plugin Name | Plugin ID | Description |
|---|---|---|
| IT Cost Management | com.snc.cost_management | Enables the definition and tracking of configuration item costs. Uses rate cards with CIs, contracts, and projects. This plugin is activated with project management v3 only. |
| Planned Task | com.snc.planned_task | Provides additional fields for tasks pertaining to time and effort as part of a planned, multi-stage process. |
| Process Flow Formatter | com.snc.process_flow_formatter | Displays the different stages in a linear process flow across the top of a record. |
| Skills Management | com.snc.skills_management | Enables an administrator to assign configured competencies, called skills, to groups or individual users. |
| Time card management | com.snc.time_card | Enables the creation of linear timelines for any activity that is bounded by two dates, such as the actual start and end dates on a project task. |
| Timeline Visualization | com.snc.timeline_visualization | Enables graphical representation of activities over time. |

# Script Includes

The following script includes are available with the v2 plugin:

| Name | Description |
|---|---|
| TimelineGanttSchedulePage | Schedule page script include for the Gantt chart. |
| TimelinePortfolioGanttPage | Schedule page script include for the portfolio Gantt chart. |
| TimelineGroupResourceSchedulePage | Schedule page script include for the group resource timeline. |
| TimelineProjectResourceSchedulePage | Schedule page script include for the project resource timeline. |

The following script includes are available with the v3 plugin.

| Name | Description |
|---|---|
| ProjectGanttSchedulePage | Schedule page script include for the Gantt chart. |
| AjaxProjectTaskUtil | Utility that calculates the end date and duration for the projects and project tasks. |
| ProjectTaskLinkUtils | Utility that provides functions for project task links. |
| PlannedTaskStateUtil | An extension to the Task State Management Utility with support for default open state and default skipped state. Configurations are defined in the task.state dictionary element, usually using dictionary overrides for planned task since state values vary by table. |
| PTLUtils | Utility functions for planned task links. |
| GetPortfolioProjectIDs | Helper function to retrieve the IDs of projects belonging to a given portfolio ID. |
| ProjectTaskManagerUtil | Utility for accessing project task manager functionality through a client callable script. Also checks if PPM attributes can be modified. |
| ProjectTemplateAjaxService | Ajax utility which is responsible for fetching project template data. |
| ProjectPortfolioUtils | Utility responsible for managing project portfolio data. |

The following script includes are available with the Fuji release.

| Name | Description |
|---|---|
| ProjectWorkbenchUtil | Workbench utility for generic functions. |
| ProjectWorkbenchAjaxService | Ajax utility which is responsible for fetching data required to draw Project Workbench as dashboard. |
| ProjectWorbenchConfig | Workbench configuration utility. |
| PPMDashboardUtil | Provides APIs for project dashboard. |
| ProjectWorkbenchAnyPhase | Abstract phase for project workbench. |
| ProjectWorkbenchService | Service which is responsible for fetching data required to draw Project Workbench as dashboard. |
| ProjectWorkbenchProjectEntity | Handles project data. |
| ProjectWorkbenchPhaseFactory | Factory for handling various project workbench phases. |
| ProjectWorkbenchMilestonePhase | Handles milestone data. |

# Client Scripts

The following client script is added with the v2 and v3 plugins.

| Script | Table | Description |
|---|---|---|
| Populate Portfolio Dashboard on Demand | Portfolio Project [pm_portfolio_project] | Populates the portfolio dashboard with values from projects in a portfolio. |

The following client scripts are added with the v3 plugin only:

| Script | Table | Description |
|---|---|---|
| Calculate Duration From End Date | Project [pm_project] and Project task [pm_project_task] | Calculates the duration of a task if you already entered an end date. |
| Calculate End Date From Duration | Project [pm_project] and Project task [pm_project_task] | Calculates the end date of a task if you already entered the duration. |
| Calculate Lag Days by Schedule | Planned Task Relationship [planned_task_rel_planned_task] | Calculates duration on changing the lag time. |
| Check Percent Complete(onChange) | Project Task [pm_project_task] | Validates percent compete value on changing percent compete. |
| Check Percent Complete Value | Project task [pm_project_task] | Validates percent compete value on submitting the form. |
| Check Valid Nested Project Start Date | Project [pm_project] | Validates the planned start date against the earliest date a nested project can start. |
| Check Valid Planned Task Start and End | Planned task [planned_task] | Validates the planned start date against the earliest start date a planned task can start. |
| Check Valid Task Start Date and Set End | Project task [pm_project_task] | Validates the planned start date against the earliest date a task can start. |
| Disable calculation type | Project [pm_project] | If calculation type is automatic and the project has children, calculation type should be disabled. |
| Hide actual start and end dates | Project [pm_project] | Hide actual start and end dates for a new form. |
| Hide the Project Task Related List | Project task [pm_project_task] | Hides the **Project Tasks** related list for the project tasks that are linked to changes. |
| Make table and link read only | Project Template Task [project_template_task] | For the record of project_template_task, makes mandatory for new record or read-only for the existing record. |
| Nested Project Hide Schedule | Project [pm_project] | Hides the **Schedule** field for nested projects. Only the top-most project can have a schedule attached to it. |
| Populate Nested Project Start Date | Project [pm_project] | Populates the start date of a nested project. |
| Populate Task Start Date | Project task [pm_project_task] | Populates the start date of a project task based on the task's dependencies and relationships. |
| Populate Planned Task Start Date | Planned task [planned_task] | Controls showing/hiding the start_date. |
| Populate Portfolio | Project [pm_project] | Sets the primary portfolio value. |

The following client scripts are added with the v3 plugin only:

| Script | Table | Description |
|---|---|---|
| WorkBenchDurationCheck | pm_project_task | Validates duration on loading the form. |
| WorkbenchWaterfallActualOrPlanned | pm_project_task | The script hides or shows the actuals depending on whether the values are present. Actuals are shown if present else planned dates are shown. For manual projects hide all the actuals. This is valid only for waterfall view. |
| WorkbenchDurationCheck | pm_project_task | Validates duration on changing the End **date** field. |
| WorkbenchUpdateCalculationType | pm_project | Updates project calculation on loading the project form. |
| WorkbenchUpdateCalculationType | pm_project_task | Workbench Update Calculation Type onLoad script on project task. |
| WorkbenchUpdateCalculationType | pm_project_task | Workbench Update Calculation Type on changing the calculation. |

# UI Policies

| UI Policy | Table | Description |
|---|---|---|
| Disable Planned Start Date for Start ASAP tasks | Project task [pm_project_task]: | The system ignores the planned start date for tasks if the **Time constraint** is set to **Start ASAP**. |
| Disable Time Constraint if Roll-up | Project Task [pm_project_task] | Disables the time_constraint for the rollup tasks. |
| Make Project Name Mandatory | Project [pm_project] | Makes the **Project Name** field mandatory on the project form. |
| Portfolio name mandatory | Portfolio [pm_portfolio] | Makes the **Portfolio name** field mandatory. |
| Disable Time Constraint if the task is a manual task or not in pending or open state or rollup task | Project task [pm_project_task] | Disables the time_constraint if the task is a manual task or rollup task or not in either the pending or open state. |
| Toggle writability of start_date depending on value in time_constraint | Project [pm_project]: | The system changes the **Planned start date** field of a project to read-only if the **Time constraint** is set to **Start ASAP**. |
| Disable Time Constraint if Roll-up | Project Task [pm_project_task] | Disables time constraint for the rollup tasks. |
| Hide Link and Rollup Fields | Project Task [pm_project_task] | Hides links and roll task fields. |
| Disable Time Constraint for the Project | Project [pm_project] | Disables the time_constraint for the project. |
| Make Project Name/Short Description mandatory | Project Task [pm_project_task] | Makes the **Name** and **Short description** mandatory. |
| Disable Time Constraint if the Task is not in Pending or Open State | Project Task [pm_project_task] | Disables the time constraint if the task is not in the pending/open state. |
| PlannedStartDateandEndDateReadonlyifSprintPresent | Project Task [pm_project_task] | Make start and end date read only if sprint is available for the project’s team. |

# UI Actions

The project management v2 and v3 plugins adds the following UI actions.

| UI Action | Tables | Description |
|---|---|---|
| Gantt Chart | planned_task, project | Displays the Gantt chart for a project. |
| Portfolio Gantt Chart | pm_portfolio | Displays the Gantt chart for a portfolio. |
| Create Baseline | pm_project | Creates a baseline for a project at the time the user initiates the UI action. |
| Copy Project | pm_project | Copies a project and all of its subtasks and relationships. |
| Copy Partial Project | pm_project, pm_project_task | Copies a project or task and its subtasks and relationships. |
| Project task creator | pm_project, pm_project_task | Adds a specified number of child tasks. |
| Redirect to top task | pm_project_task | Shows the parent task above the current task. |
| Refresh project | pm_portfolio_project | Refreshes the portfolio project. |
| Refresh portfolio | pm_portfolio | Refreshes the portfolio. |
| Create portfolio | pm_portfolio | Creates the portfolio. |
| Start project | pm_project | Starts a project and automatically changes the project state to **Work in Progress**. |
| Resource timeline (v2 only) | pm_project | Displays the resource timeline tool (v2 only). |
| Create change and link from project task | pm_project_task | Creates a change request record and links it to the project task. |
| Link existing change to project task | pm_project_task | Links a change request record to the project task. |
| Export to XML | pm_project | Exports the project and its tasks to be imported into another instance. |
| Create Project | incident, problem, change_request | Creates a project out of an incident, problem, or change request. |
| Create Project | Project [pm_project] | Creates a project. |
| Project Workbench | Project [pm_project] | Saves user preference for the last accessed workbench. |
| Submit | Project Task [pm_project_task] | Saves a new record and redirects back to previous screen (usually a list). |
| Submit | Project [pm_project] | Saves a new record and redirects back to previous screen (usually a list). |

# Business Rules

## Project Management v2

The project management v2 plugin adds the following business rules.

| Business Rule | Table | Description |
|---|---|---|
| Project Cost Rollup | Planned task [planned_task] | Calculates and rolls up project cost. This is installed with the Project Management plugin but also works with the Project Management Costing Add-on plugin, which must be activated separately. |
| Process Project Task State Changes | Project [pm_project] | When the project record **State** is modified, this business rule starts child and successor tasks. |
| Process Project Task State Changes | Project Task [pm_project_task] | When the project task record state is modified, this business rule starts child and successor tasks. |

## Project Management v3

All business rules for project management v3 run on project management tables, such as Project [pm_project] and Project Task [pm_project_task]. No project management v3 business rules run on the Planned task [planned_task] table. If you created or modified any project management v2 business rules to run on Planned task [planned_task], you must modify the business rules to run on a project management table.

The v3 plugin adds the following business rules.

| Business Rule | Table | Description |
|---|---|---|
| Process State Change | Project [pm_project] and Project Task [pm_project_task] | Processes related tasks when a project record state is modified. This replaces the Process Project State Change and Process Project Task State Changes business rules from the v2 plugin. |
| Update Project Link | Change Request [change_request] | Links change requests to project tasks. |
| Unlink Parent | Project Task Link [pm_project_task_link] | Removes the link between a project task and its linked change request. |
| Recalculate | Project [pm_project] and Project Task [pm_project_task] | Updates the project task link's schedule when modifications are made to the linked change request. |
| Recalculate after Delete | Project [pm_project] | Recalculates project values when a task is deleted. |
| Calculate ROI | Project [pm_project] | Calculates the appropriate **ROI%** value for a project. |

The following business rules ran on the Planned task [planned_task] table for project management v2. With project management v3, these now run on one or more of the project management tables.

| Business Rule | Table | Description |
|---|---|---|
| Delete children | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] | Removes child records. |
| Set Actual Work Start Value | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] | Sets the the work start date. |
| Set Close Data on Inactive | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] | Sets the closed date on project and task records that are inactive. |
| Auto close milestones | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] | Closes milestone tasks. |
| planned task global events | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] | Triggers most of the common planned task events. |
| Update Parent's Percent Complete | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] | Updates a parent task's completion percentage when child tasks are updated. |

The v3 plugin deletes the following business rule.

| Business Rule | Table | Description |
|---|---|---|
| Prevent duplicate/recursive relationship | Planned Task Relationship [planned_task_rel_planned_task] | Prevents recursive relationships between tasks. |

The following business rules are available starting with the Eureka release. All of these business rules make sure that fields on non-project tasks are populated correctly when these tasks are added to a project.

| Business Rule | Table |
|---|---|
| Initialize Node on Planned Task | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] |
| Set top task with version | Planned task [planned_task] |
| Set top task on children with version | Planned task [planned_task] |
| Change Parent | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] |
| Uninitialize Node | Project [pm_project], Planned task [planned_task], and Project task [pm_project_task] |

The following business rules are available starting with the Fuji release.

| Business Rule | Table | Definition |
|---|---|---|
| DeleteVTBBoardonPhaseDeletion | Project Task [pm_project_task] | Deletes the visual task board on deleting the corresponding phase. |
| UpdateTestPlanForTestPhaseOnDelete | Project Task [pm_project_task] | Updates the test plan on deleting the test phase. |
| ProjectWorkbenchPhaseValidationAndUpdate | Project Task [pm_project_task] | Validates a phase before updating a phase. |
| ProjectWorkbenchPhaseValidationAndUpdate | Project Task [pm_project_task] | Validates a phase and project before updating a phase. |
| UpdatePhaseInStories | Project Task [pm_project_task] | Updates the phase column in the story. |
| WorkbenchUpdateCalculationType | Project [pm_project] | Generates messages if validation fails. |

# Schedule Pages

The project management v2 and v3 plugins adds the following schedule pages.

| Schedule Page | Description |
|---|---|
| Gantt Chart (v2) or Project Gantt Chart (v3) | Displays all of a project's tasks in a Gantt chart. |
| Portfolio Gantt Chart | Displays all of a portfolio's projects in a Gantt chart. |
| Group Resource Timeline (v2 only) | Displays all planned tasks for users who belong to the specified group. |
| Project Resource Timeline (v2 only) | Displays all planned tasks for resources who belong to the specified project (v2 only). For v3, this schedule page has been moved to the Resource Management plugin. |

# Schedules

The Project Management v3 plugin adds the following schedule to the Schedule [cmn_schedule] table.

| Schedules | Description |
|---|---|
| Project Management Schedule | The default schedule for all new projects, which is Monday to Friday 8 A,M. to 12 P.M. and 1 P.M. to 5 P.M. (for a total of 8 hours), not including weekends. |
| Default MS Project | The default project import schedule. |

# Properties

The Project Management v2 and v3 plugins add the following properties. Access the properties by navigating to **Project > Administration > Properties**.

| Property Name | Description | Default |
|---|---|---|
| com.snc.project.calculate_roi | Calculate ROI percentage based on a project's estimated cost and its net value. | Yes |
| com.snc.project.rollup.cost | Enable project cost rollup (estimated and actual) - updating the cost of a project task will update the cost of its parent. | No |
| com.snc.project.auto_close_milestones | Automatically close project milestone tasks when they change to work state. | No |
| com.snc.project.default_schedule | Stores the sys_id if the default schedule attached to projects. (Available starting with v3.) | Yes |
| com.snc.project.loglevel | Enables debugging of new project scripts. (Available starting with v3.) | 0 (debugging disabled) |
| com.snc.project.wbs_gantt | Adds the numbers that represent the order of tasks in the project hierarchy as seen on the work breakdown structure. (Available starting with v3.) | No |
| com.snc.project.gantt.exclude_list | Excludes the entity types (table names) included in the comma-delimited list from the project Gantt chart. | dmn_decision, dmn_requirement |
| com.snc.resource_management.percentage_allocation_normal | If resource_management reporting for a resource/group has value of percentage_allocation less than this value the color will be shown as green. | 50 |
| com.snc.resource_management.percentage_allocation_warning | If resource_management reporting for a resource/group has value of percentage_allocation less than this value and greater than the com.snc.resource_management.percentage_allocation_normal, then color will be shown as orange. | 90 |
| glide.cost_mgmt.process_task_top_task | When creating a task expense line, should the system also create expense lines for the task's top task. For example, project is a project task's top task. | True |

## User Roles

The project management v2 and v3 plugins adds the following user roles.

| Role | Contains Roles | Description |
|---|---|---|
| project_portfolio_user | none | Can view portfolio information but cannot modify it. |
| project_user | project portfolio user | Can only modify a few Project form fields, such as journal fields. Can also modify additional fields on the Project Task form, such as **Time constraint** and **State**. |
| project_manager | project user<br>project portfolio user<br>resource_user<br>view_changer | Has configuration access right to all Project application features and functionality. (See Resource Management for more information about the resource_user role). |

# Project Management v2 to v3 Upgrade

## Overview

Project management v3 provides a variety of enhancements to the Project application. Upgrading from project management v2 to v3 renames or deactivates several customer-modified v2 business rules, scripts, UI actions, and other components. Additionally, the upgrade changes the Planned Task [planned_task] table.

Project management v3 is available starting with the Dublin release. Starting with the Eureka release, only the v3 plugin is available.

## Planned Task Changes

Upgrading adds a column to the Planned Task table: **Version**. This column enables business rules on project management v3 to run on the Project or Project Task tables, rather than the Planned Task table.

### Business Rules

All business rules for project management v3 run on project management tables, such as Project [pm_project] and Project Task [pm_project_task]. No project management v3 business rules run on the Planned Task table. If you created or modified any project management v2 business rules to run on the Planned Task table, you must modify the business rules to run one of the project management tables.

# The Upgrade Process

The upgrade process involves activating the Project Management v3 plugin, checking which components were deactivated or renamed, reviewing migration logs, and reactivating any necessary components.

## Performing the Upgrade

To upgrade to v3, activate the Project Management (`com.snc.project_management_v3`) plugin.

**Click the plus to expand instructions for activating a plugin.**

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

## Deactivation and Renaming

The upgrade process deactivates or renames components of the v2 plugin that you modified or created. Renaming involves appending the word **OLD** to the previous name to prevent any conflicts with v3 components. Deactivated components can be reactivated after upgrade.

The exact action taken depends on whether it is an existing component that you modified or a new component that you created.

| Component | Existing Component Modified | New Component Created |
|---|---|---|
| Business rules | Rename and deactivate | Deactivate |
| UI policies | Rename and deactivate | Deactivate |
| UI actions | Rename and deactivate (most) | Deactivate |
| Script include | Rename and deactivate | No action taken |
| UI pages | Rename | No action taken |
| Client scripts | No action taken | deactivated |
| Data dictionary and tables | No action taken | No action taken<br>**Note:** Any deleted tables are reinstalled. |
| Modules | No action taken | No action taken |
| Schedule pages<br>**Note:** Default schedule pages are reinstalled. | Rename | No action taken |

The following components, created by an administrator, remain after upgrade:

- Email notifications
- Access control lists (ACLs)
- Data policies
- Field styles
- View rules

# Post-Upgrade Steps

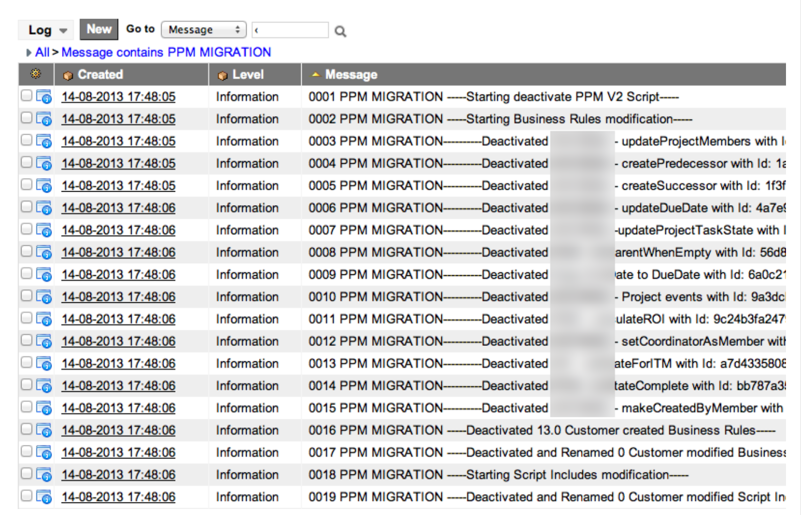
Once migration is complete, view the following system logs for detailed information.

- **PPM migration:** shows any customizations, such as business rules, client scripts, and UI pages, that should be recreated manually.
- **Project migration:** shows all existing project records and task records that were migrated.

To view PPM migration logs:

1. Navigate to **System Log > All**.
2. Edit the filter to search for **[Message] [contains] [PPM MIGRATION]**.
3. In the filtered log messages, view the migration details.

   The log also lists customizations that were not migrated automatically and should be recreated manually. Components that are commonly customized include client scripts, script includes, and processors.



All > Message contains PPM MIGRATION

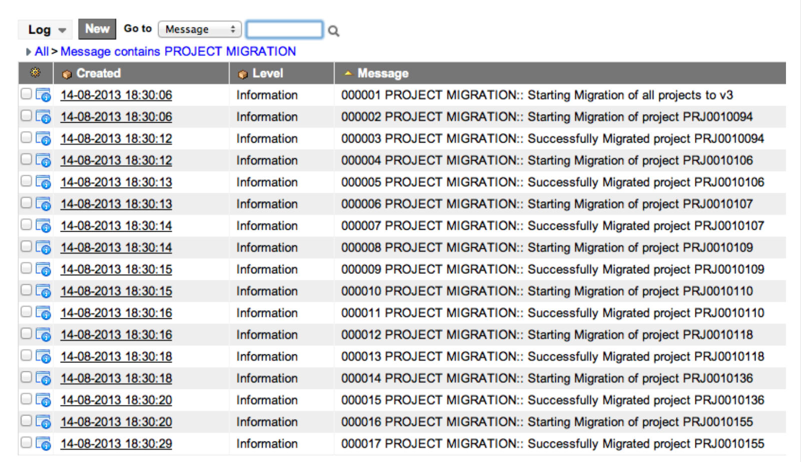
| | Created | Level | Message |
|---|---|---|---|
| ☐ | 14-08-2013 17:48:05 | Information | 0001 PPM MIGRATION -----Starting deactivate PPM V2 Script----- |
| ☐ | 14-08-2013 17:48:05 | Information | 0002 PPM MIGRATION -----Starting Business Rules modification----- |
| ☐ | 14-08-2013 17:48:05 | Information | 0003 PPM MIGRATION---------Deactivated [redacted] - updateProjectMembers with I |
| ☐ | 14-08-2013 17:48:05 | Information | 0004 PPM MIGRATION---------Deactivated [redacted] - createPredecessor with Id: 1a |
| ☐ | 14-08-2013 17:48:05 | Information | 0005 PPM MIGRATION---------Deactivated [redacted] - createSuccessor with Id: 1f3f |
| ☐ | 14-08-2013 17:48:06 | Information | 0006 PPM MIGRATION---------Deactivated [redacted] - updateDueDate with Id: 4a7e |
| ☐ | 14-08-2013 17:48:06 | Information | 0007 PPM MIGRATION---------Deactivated [redacted] -updateProjectTaskState with I |
| ☐ | 14-08-2013 17:48:06 | Information | 0008 PPM MIGRATION---------Deactivated [redacted] arentWhenEmpty with Id: 56d8 |
| ☐ | 14-08-2013 17:48:06 | Information | 0009 PPM MIGRATION---------Deactivated [redacted] ate to DueDate with Id: 6a0c2 |
| ☐ | 14-08-2013 17:48:06 | Information | 0010 PPM MIGRATION---------Deactivated [redacted] - Project events with Id: 9a3dc |
| ☐ | 14-08-2013 17:48:06 | Information | 0011 PPM MIGRATION---------Deactivated [redacted] ulateROI with Id: 9c24b3fa247 |
| ☐ | 14-08-2013 17:48:06 | Information | 0012 PPM MIGRATION---------Deactivated [redacted] - setCoordinatorAsMember wit |
| ☐ | 14-08-2013 17:48:06 | Information | 0013 PPM MIGRATION---------Deactivated [redacted] ateForITM with Id: a7d433580 |
| ☐ | 14-08-2013 17:48:06 | Information | 0014 PPM MIGRATION---------Deactivated [redacted] tateComplete with Id: bb787a3 |
| ☐ | 14-08-2013 17:48:06 | Information | 0015 PPM MIGRATION---------Deactivated [redacted] - makeCreatedByMember with |
| ☐ | 14-08-2013 17:48:06 | Information | 0016 PPM MIGRATION -----Deactivated 13.0 Customer created Business Rules----- |
| ☐ | 14-08-2013 17:48:06 | Information | 0017 PPM MIGRATION -----Deactivated and Renamed 0 Customer modified Business |
| ☐ | 14-08-2013 17:48:06 | Information | 0018 PPM MIGRATION -----Starting Script Includes modification----- |
| ☐ | 14-08-2013 17:48:06 | Information | 0019 PPM MIGRATION -----Deactivated and Renamed 0 Customer modified Script In |

Example PPM migration log records

To view project migration logs:

1. Navigate to **System Log > All**.
2. Edit the filter to search for **[Message] [contains] [PROJECT MIGRATION]**.
3. In the filtered log messages, view the migration details.

   The log shows each project record that was migrated.
4. Use the **Go to** search box to find the tasks and projects.



All > Message contains PROJECT MIGRATION

| | Created | Level | Message |
|---|---|---|---|
| ☐ | 14-08-2013 18:30:06 | Information | 000001 PROJECT MIGRATION:: Starting Migration of all projects to v3 |
| ☐ | 14-08-2013 18:30:06 | Information | 000002 PROJECT MIGRATION:: Starting Migration of project PRJ0010094 |
| ☐ | 14-08-2013 18:30:12 | Information | 000003 PROJECT MIGRATION:: Successfully Migrated project PRJ0010094 |
| ☐ | 14-08-2013 18:30:12 | Information | 000004 PROJECT MIGRATION:: Starting Migration of project PRJ0010106 |
| ☐ | 14-08-2013 18:30:13 | Information | 000005 PROJECT MIGRATION:: Successfully Migrated project PRJ0010106 |
| ☐ | 14-08-2013 18:30:13 | Information | 000006 PROJECT MIGRATION:: Starting Migration of project PRJ0010107 |
| ☐ | 14-08-2013 18:30:14 | Information | 000007 PROJECT MIGRATION:: Successfully Migrated project PRJ0010107 |
| ☐ | 14-08-2013 18:30:14 | Information | 000008 PROJECT MIGRATION:: Starting Migration of project PRJ0010109 |
| ☐ | 14-08-2013 18:30:15 | Information | 000009 PROJECT MIGRATION:: Successfully Migrated project PRJ0010109 |
| ☐ | 14-08-2013 18:30:15 | Information | 000010 PROJECT MIGRATION:: Starting Migration of project PRJ0010110 |
| ☐ | 14-08-2013 18:30:16 | Information | 000011 PROJECT MIGRATION:: Successfully Migrated project PRJ0010110 |
| ☐ | 14-08-2013 18:30:16 | Information | 000012 PROJECT MIGRATION:: Starting Migration of project PRJ0010118 |
| ☐ | 14-08-2013 18:30:18 | Information | 000013 PROJECT MIGRATION:: Successfully Migrated project PRJ0010118 |
| ☐ | 14-08-2013 18:30:18 | Information | 000014 PROJECT MIGRATION:: Starting Migration of project PRJ0010136 |
| ☐ | 14-08-2013 18:30:20 | Information | 000015 PROJECT MIGRATION:: Successfully Migrated project PRJ0010136 |
| ☐ | 14-08-2013 18:30:20 | Information | 000016 PROJECT MIGRATION:: Starting Migration of project PRJ0010155 |
| ☐ | 14-08-2013 18:30:29 | Information | 000017 PROJECT MIGRATION:: Successfully Migrated project PRJ0010155 |

Example project migration log records

## Reactivating Business Rules, UI Actions, and UI Policies

Administrators can reactivate business rules, UI actions, and UI policies manually one by one. To reactivate them in bulk, make a request to ServiceNow Technical Support.

# Project Management v1 to v2 Upgrade

## Overview

If you are currently using Project Management v1, and intend to upgrade to Project Management v2, *read this entire article*. Version 1 customers upgrading to v2 must perform some initial analysis *prior* to loading the v2 plugin.

## Who does this apply to?

When you upgrade from Project Management v1 to v2, the system makes many changes to remove all the old, v1 functionality, and make way for the new v2 files. We want to ensure that no customizations in the system get overwritten in this process. Contact ServiceNow Technical Support *prior* to loading the v2 plugin if you are a v1 customer and you have:

- Modified any of the v1 Business Rules, Security Rules, UI Policies, Client Scripts, forms, or lists.
- Added your own Business Rules, Security Rules, UI Policies, Client Scripts to the *pm_project* or *pm_project_task* tables.
- Data in the *pm_project* or *pm_project_*task tables that is not accounted for in the *Data capture* section below (basically if you have added custom fields and want that data moved to the new system).

## Worst case

Forms/Lists - If you've modified any pm_project or pm_project_task forms or lists, we won't be able to overwrite them with our new forms and lists (as you now own them), so after the upgrade they'll look as if they're missing some of their fields. This is due to us dropping most of the older fields, and replacing them with new ones. Fixing this is as simple as configuring the forms and lists again and re-adding the appropriate fields.

Data - When you upgrade your projects will be converted to the new v2 system. But the data in any custom fields might be left behind. Recovery of this data is simple, but is something that's not preferable if we can get ahead of it.

Policy - If you've made any changes to the out-of-box business rules, or have made any of your own policy, it may conflict with the new v2 policy.

If you find that any of these might apply to you, please contact ServiceNow Technical Support *prior* to loading the v2 plugin.

## The upgrade process

### Data Capture

Prior to making any changes to your system, we capture all existing projects (pm_project) and project tasks (pm_project_task) for the later step of reinserting all the data into the new structures. We capture the following fields:

- name (**NOTE**: short_description is NOT captured - rather, the name field is used to populate short_description. If a customer is using short_description in Project Management v1, they must copy those values to the name field BEFORE upgrading to v2)
- order
- state
- description

- cmdb_ci
- url
- work_notes
- comments
- location
- priority
- impact
- urgency
- number
- estimated_cost
- project_lead
- assigned_to
- assignment_group
- planned_start_date
- planned_end_date
- planned_duration
- parent
- project
- active
- percent_complete
- actual_start_date
- actual_end_date
- actual_duration
- automate
- company

**Note:** *If you have added custom fields to the task, **pm_project**, or **pm_project_task** tables, the data is* not *captured and is therefore* not *reinserted after the upgrade. If this occurs, contact ServiceNow Technical Support* prior *to loading the v2 plugin. We will help you export the custom data and reimport it after the upgrade.*

## Fields removed

After capturing all the appropriate data, the following fields from the original v1 plugin are no longer used and are removed from the system:

- pm_project - name
- pm_project - automate
- pm_project - template
- pm_project - task_count
- pm_project - planned_start_date
- pm_project - planned_end_date
- pm_project - planned_duration
- pm_project - actual_start_date
- pm_project - actual_end_date
- pm_project - actual_duration
- pm_project - estimated_cost
- pm_project - project_lead
- pm_project - template

- pm_project_task - name
- pm_project_task - has_child
- pm_project_task - project
- pm_project_task - uid
- pm_project_task - planned_start_date
- pm_project_task - planned_end_date
- pm_project_task - planned_duration
- pm_project_task - actual_start_date
- pm_project_task - actual_end_date
- pm_project_task - actual_duration
- pm_project_task - estimated_cost
- task - order

## Policy Removed

All v1 Business Rules, UI Policies, Security Rules, Client Scripts, and so on are removed as part of the upgrade. With the upgrade, you get an entirely new set of rules that are better designed to work with v2.

If you have customized any of the v1 Business Rules, or created any of your own rules against the *pm_project* or *pm_project_task* tables, these rules must be removed manually (to prevent them from interrupting any of the new functionality).

## Data Reinserted

After capturing your data, removing the old fields, and removing the old policy, we insert the new v2 fields and policy. Once that is complete, we reinsert your data. We reinsert most of the attributes listed above into the same fields. We also convert some of your old data using the following rules:

- Name becomes Short description
- Order gets converted to relationships in the planned_task_rel_planned_task table
- Documentation becomes HTML description
- Estimated cost (estimated_cost) becomes Estimated cost (cost)
- Project lead becomes Assigned to (only for project records)
- Planned start date (planned_start_date) becomes Planned start date (start_date)
- Planned end date (planned_end_date) becomes Planned end date (end_date)
- Planned duration (planned_duration) becomes Planned duration (duration)
- Project becomes Top task
- Actual start date (actual_start_date) becomes Actual start date (work_start)
- Actual end date (actual_end_date) becomes Actual start date (work_end)
- Actual duration (actual_duration) becomes Actual duration (work_duration)
- Automate gets converted to all project tasks in the form of their Time constraint...'ASAP' if automate was checked, 'Start on' if not

# What should I do?

The best way to be sure that your upgrade is guaranteed to go smoothly is to contact Technical Support and let us do a quick analysis of your system. We'll be able to tell you whether additional work will be necessary in order to complete your upgrade. But if you'd like to do it on your own, follow these steps:

## Data backup

If you think you're ok, but want a fail-safe, go ahead and do the following:

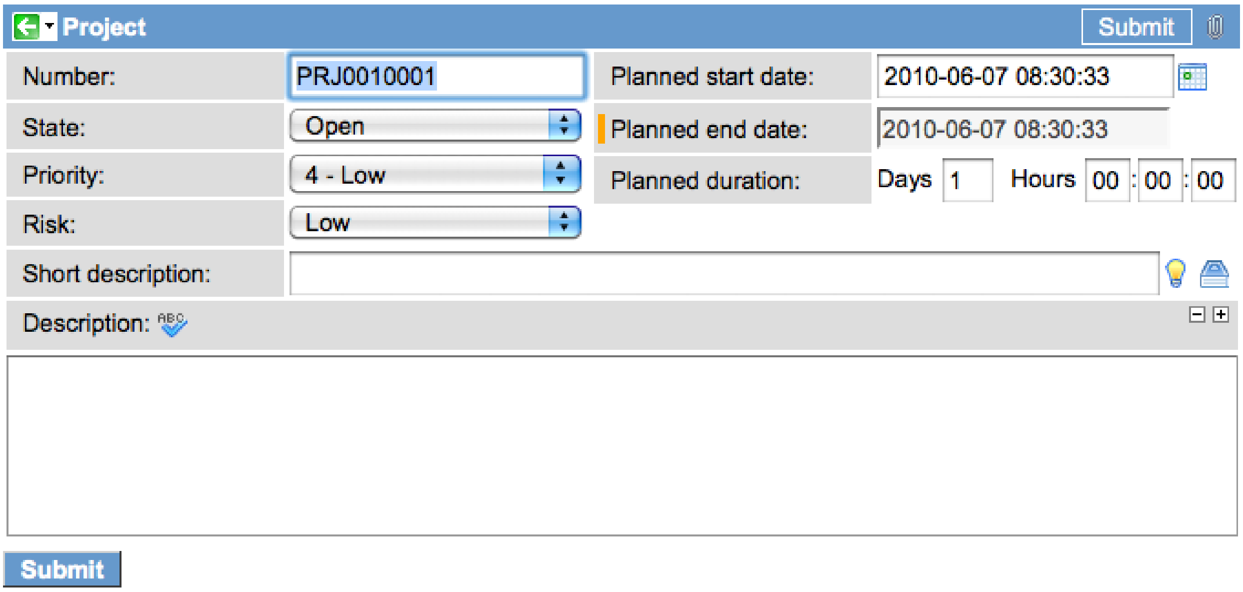
- Create a report against the 'Project (pm_project)' table. Make it a list report, not grouped, selecting ALL columns to be displayed.
- Run the report, and when the results come back, right-click on the header bar of the list, and choose 'Export -> Excel'.
- Now do the exact same process for the 'Project task (pm_project_task)' table.

Taking this rudimentary data backup ensures that we can quickly recover if any data is lost during the upgrade.

## Reconfigure forms and lists

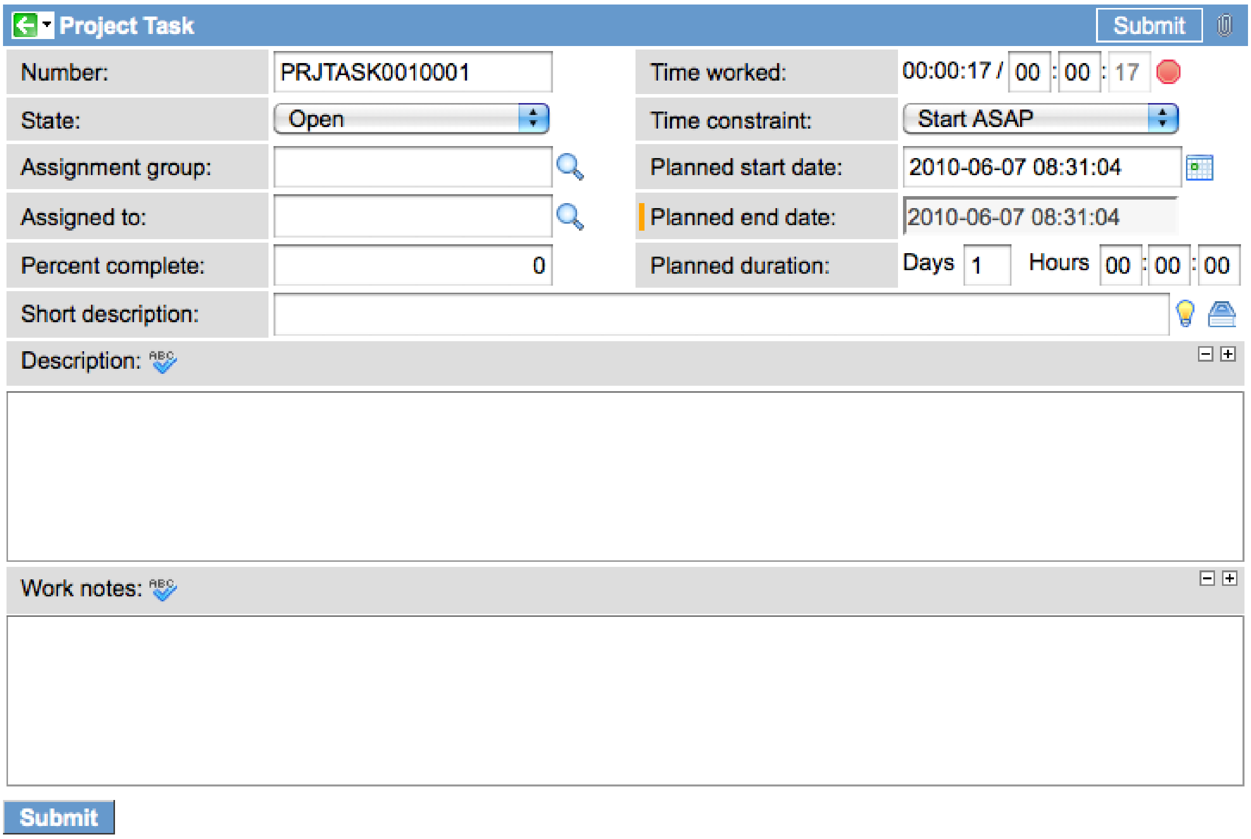
If you had made changes to them while using the original v1 plugin, your forms and lists may look odd after your upgrade. Not to worry, they just need to be reset using our new fields. Feel free to configure them any way you'd like, but here are the defaults:

### Project form (default view)



| | | | |
|---|---|---|---|
| Number: | PRJ0010001 | Planned start date: | 2010-06-07 08:30:33 |
| State: | Open | Planned end date: | 2010-06-07 08:30:33 |
| Priority: | 4 - Low | Planned duration: | Days 1 Hours 00 : 00 : 00 |
| Risk: | Low | | |
| Short description: | | | |
| Description: | | | |

## Project task form (default view)



Project Task Submit

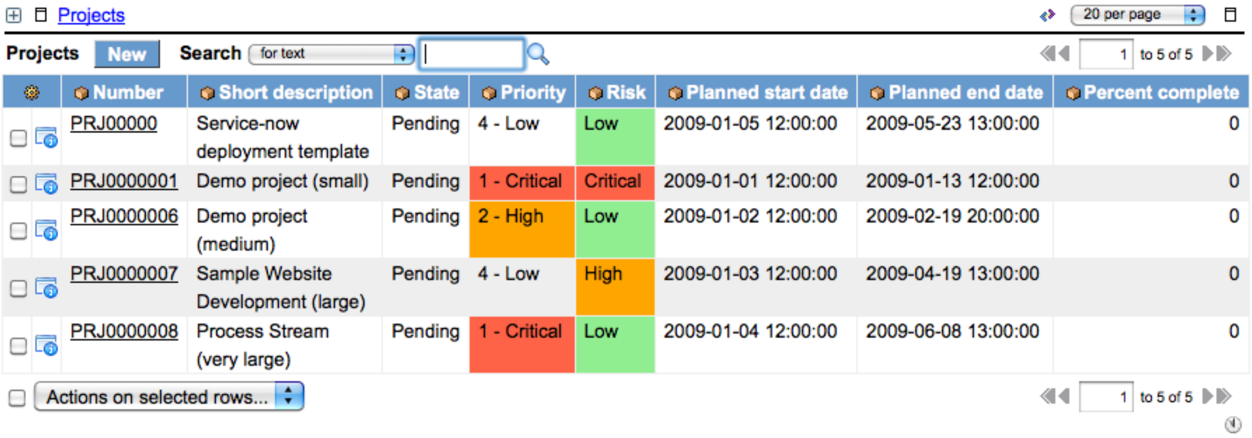
| | | | |
|---|---|---|---|
| Number: | PRJTASK0010001 | Time worked: | 00:00:17 / 00 : 00 : 17 |
| State: | Open | Time constraint: | Start ASAP |
| Assignment group: | | Planned start date: | 2010-06-07 08:31:04 |
| Assigned to: | | Planned end date: | 2010-06-07 08:31:04 |
| Percent complete: | 0 | Planned duration: | Days 1 Hours 00 : 00 : 00 |
| Short description: | | | |

Description:

Work notes:

Submit

## Project list (default view)



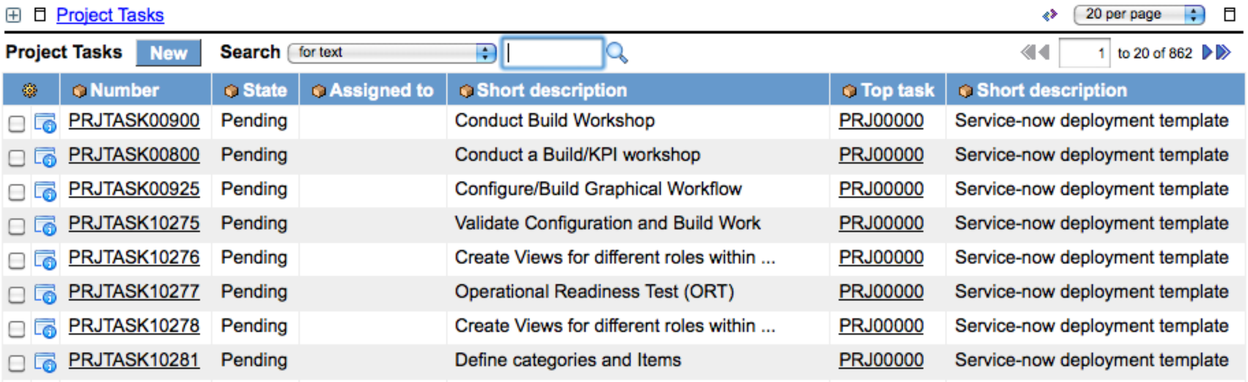
Projects

Projects New Search for text 20 per page 1 to 5 of 5

| Number | Short description | State | Priority | Risk | Planned start date | Planned end date | Percent complete |
|---|---|---|---|---|---|---|---|
| PRJ00000 | Service-now deployment template | Pending | 4 - Low | Low | 2009-01-05 12:00:00 | 2009-05-23 13:00:00 | 0 |
| PRJ0000001 | Demo project (small) | Pending | 1 - Critical | Critical | 2009-01-01 12:00:00 | 2009-01-13 12:00:00 | 0 |
| PRJ0000006 | Demo project (medium) | Pending | 2 - High | Low | 2009-01-02 12:00:00 | 2009-02-19 20:00:00 | 0 |
| PRJ0000007 | Sample Website Development (large) | Pending | 4 - Low | High | 2009-01-03 12:00:00 | 2009-04-19 13:00:00 | 0 |
| PRJ0000008 | Process Stream (very large) | Pending | 1 - Critical | Low | 2009-01-04 12:00:00 | 2009-06-08 13:00:00 | 0 |

Actions on selected rows... 1 to 5 of 5

## Project task list (default view)



Project Tasks

Project Tasks New Search for text 20 per page 1 to 20 of 862

| Number | State | Assigned to | Short description | Top task | Short description |
|---|---|---|---|---|---|
| PRJTASK00900 | Pending | | Conduct Build Workshop | PRJ00000 | Service-now deployment template |
| PRJTASK00800 | Pending | | Conduct a Build/KPI workshop | PRJ00000 | Service-now deployment template |
| PRJTASK00925 | Pending | | Configure/Build Graphical Workflow | PRJ00000 | Service-now deployment template |
| PRJTASK10275 | Pending | | Validate Configuration and Build Work | PRJ00000 | Service-now deployment template |
| PRJTASK10276 | Pending | | Create Views for different roles within ... | PRJ00000 | Service-now deployment template |
| PRJTASK10277 | Pending | | Operational Readiness Test (ORT) | PRJ00000 | Service-now deployment template |
| PRJTASK10278 | Pending | | Create Views for different roles within ... | PRJ00000 | Service-now deployment template |
| PRJTASK10281 | Pending | | Define categories and Items | PRJ00000 | Service-now deployment template |

# Setting up a Project

## Proposing a New Project

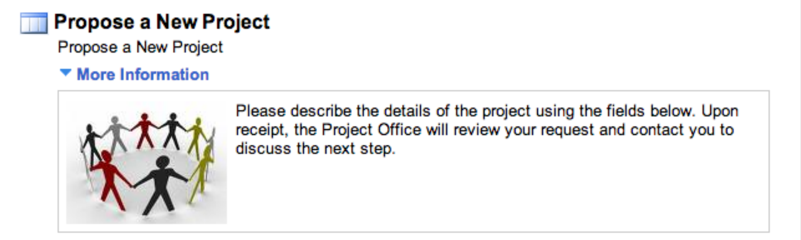
### Overview

Service Catalog administrators can customize the self-service interface to include a link for users to request a project. The link is created through a project record producer, which is included with the Project Management application and is active by default.

### Submitting a Project Request

To have Service Catalog users submit a project request:

1. Navigate to **Self-Service > Service Catalog**.
2. Click the **Can we help you?** link.
3. Click **Propose a New Project**.



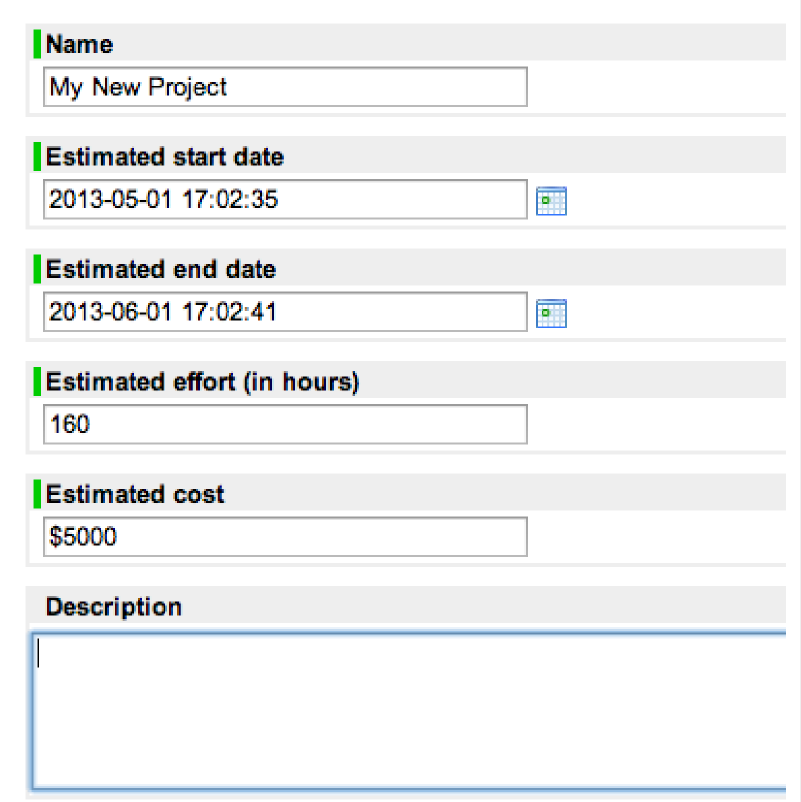
Propose a New Project

Propose a New Project

Please describe the details of the project using the fields below. Upon receipt, the Project Office will review your request and contact you to discuss the next step.

Proposing a New Project

4. Fill out the project form.



Name
My New Project

Estimated start date
2013-05-01 17:02:35

Estimated end date
2013-06-01 17:02:41

Estimated effort (in hours)
160

Estimated cost
$5000

Description

Proposing a New Project

By default, the project form is populated as follows:

- **Name** field on the submission form becomes the **Short description** on the project form
- **Estimated start date** becomes the **Planned start date**
- **Estimated cost** remains as such and is visible on the project form if the view is set to **Advanced**.

## Modifying the Project Record Producer

Administrators can make modifications to the project record producer, such as changing which service catalog category it belongs to, or making it more prominent on the Self Service interface.

1. Navigate to **Service Catalog > Catalog Definition > Record Producers**.
2. In the list of record producers, click **Propose a New Project**.
3. Modify the record as needed. Useful modifications include:

- **Category:** select a different category other than **Can We Help You?** or create a new category for project management.
- **Picture:** update the picture with your company logo.
- **Variables:** add, remove, or modify the fields that appear on the request form. The following are helpful variables to add:
  - **Configuration item:** the software or service affected. Make this variable a **Reference** type on the Configuration Item [cmdb_ci] table.
  - **Business case:** the case or justification for the project. Make this variable a **Mutli Line Text** type with the name **business_case**, which is the column name on the Project table for the **Business case** field on the Project form.

The fields on the submission form can map to the fields on the Project form. This happens because the variables that you create are associated with the Propose a New Project catalog item, which by default is associated with the Project [pm_project] table. The name of the variable on the record producer must match the column name on the Project table for the field you want to map to.

See Record Producer for more information about creating and modifying record producers.

# Creating a Project

**Note:** *This article applies to Fuji. For more current information, see Create a Project* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

## Overview

Creating the project is the first step in the Project Management process. Define important aspects of the project such as duration, estimated cost, and the net value to the organization.

To create and edit portfolios, projects, and tasks, users must have the project_manager role in their user profile record. See Installed with Project Management for more information on roles in the Project application.

### Key Concepts and Terms for This Topic

- **Time constraint:** a restriction on a project task that determines when it should start. Two options are available: **Start ASAP** and **Start on specific date**. Note that the time constraint for a task might change based on dependencies and relationships.
- **Planned and actual values:** planned values (and estimated values for costs) are entered manually and used before a project or task begins. ServiceNow calculates rollups for planned items automatically. The value for **Actual start date** can be entered manually, but other actual values are determined by child tasks or the calculation of time cards or expense lines.
- **Advanced view:** used to view cost fields, such as **Estimated cost** and **ROI%**.
- **Project [pm_project] table:** the table where project records are saved. This table extends the Planned task [planned_task] table.

## Creating the Project

Create a ServiceNow project using one of the following methods:

- Create a new project in the Project application and configure it manually.
- Copy an existing project.
- Create a project from the project workbench.
- Create a project from the demand workbench.
- Create a project from another task record, such as an incident, problem, or change request.
- Import a Microsoft Project file into ServiceNow project management.

# Creating a Project Manually

To create a new project:

1. Navigate to **Project > Projects > Create New**.

   The Project form appears with some pre-populated data and in the last view selected. Project forms have the following views:

   - **Default view:** shows a basic set of form fields needed to start a project.
   - **Advanced:** shows cost-related fields and other special fields like **Actual effort** and **Actual cost**.
   - **WBS:** shows the WBS List related list.
2. Fill out the Project form (see table).
3. Click **Submit**.

## Project Form Fields for the Fuji Release

The following table describes the project form fields from all views.

| Field (Default View) | Description |
|---|---|
| Project Name | [Required] The name of the project. |
| Project manager | The project manager assigned to this project. |
| Portfolio | The primary portfolio to which this project belongs. A project can belong to multiple portfolios. Click the magnifying glass icon to select a portfolio from the choice list. Mouse over the information icon to display a popup window with the portfolio details. Click the information icon to display the portfolio page.<br>**Note:** If a Project view is added to a portfolio from the Portfolio form, the **Portfolio** field is populated with the portfolio name, if there is no primary portfolio currently assigned. If a portfolio is deleted, the portfolio name is removed from the **Portfolio** field on the Project form. |
| Business service | The configuration item (CI) affected by this project. |
| Number | A system generated with a configurable prefix. See Managing Record Numbering for more information about customizing number prefixes. |
| State | The current state of the project. All new projects begin as **Pending** and automatically change to **Work in progress** when you click the **Start project** button on the Project form.<br>Following are the default states: **Pending**, **Open**, **Work in Progress**, **Closed Complete**, **Closed Incomplete**, **Closed Skipped**. |
| Percent complete | The percentage of the project that has been completed. |
| Calculation | The type of calculation to use for task dependencies:<br>• **Manual:** parent task dates do not reflect any changes made to dependents or child tasks.<br>• **Automatic:** parent task dates are automatically updated to reflect any changes made to dependents or child tasks.<br>For more information, see Calculating Project Task Dependencies. |
| Schedule | The work schedule to be used for this project:<br>• **v3 application:** the default schedule is an 8-hour work day (from 8 to 12 and 1 to 5). A day is considered as a working day, not a 24-hour day.<br>• **v2 application:** the optional schedule shows when project team members work on the project. Select a predefined schedule or create a new one on the Schedules table. The system uses the schedule to calculate the **Planned end date**. If no schedule is selected, the system calculates a planned duration of one day as 24 work hours. |

| | |
|---|---|
| Planned start date | The intended date the project should begin. This value is set to the earliest time that the project schedule allows (v3 application). For example, if the project task is created at 3 P.M. and the default schedule is in use (which has an 8 A.M. start date), the default task start is 8 A.M. the next day. ***Note:*** The planned start date must be within 15 years of the current date.<br>Click the calendar icon and select a date to start this project. Projects do not automatically start on the planned start date. The project actually starts when you click **Start project** on the Project form. |
| Planned end date | The intended date the project should end. ***Note:*** The planned end date must be within 15 years of the current date.<br>• **v3 application:** This field is editable but is automatically updated when the duration changes or when the planned start date changes (the planned duration is used to calculate the new end date).<br>• **v2 application:** [Read-only] The intended date the project should finish. The system calculates this field from the **Planned start date**, the **Planned duration**, and the **Schedule** (if selected). |
| Planned duration | The expected duration of this project. ***Note:*** The number of days cannot exceed 1500.<br>• **v3 application:** The duration is recalculated if the planned end date changes. The duration also considers the project schedule, accounting for any non-work time in the schedule. For example, if the default schedule is used, with a standard 8-hour work day, a project that starts at 8 A.M. on July 1 and ends at noon on July 2 is calculated as 1 day and 4 hours, not 28 hours. Any project or project task with no children is restricted to a maximum duration of 1500 days.<br>• **v2 application:** Enter the value in days, hours, minutes, or seconds. If tasks are present, this field is read-only, and the duration is calculated from the planned duration of the child tasks. The estimated duration shows total project time and takes the project schedule into consideration. |
| Planned effort | An estimate of how much time this project will take to complete. This is a rollup calculation that sums planned effort values for all tasks in this project. If no tasks are in the project, this field is editable. If tasks are configured, this field becomes a read-only rollup calculation and overwrites any earlier entry that you made. |
| Actual start date | The date that this project actually began. If the project has not been started, this field is modifiable. At the time the project starts, this field becomes read only. |
| Actual end date | The date that this project actually ended. If the project has not been started, this field is modifiable. At the time the project completes, this field becomes read only. |
| Actual duration | The actual duration of the project from project start to project closure. As with planned duration, the actual duration shows total project time and takes the project schedule into consideration. |
| Actual effort | The actual number of hours charged to the resources on this project. If you are using the Time Cards application, ServiceNow automatically calculates the value for this field using the totals for the time worked from the time cards of all the resources who worked on this project. |
| Estimated cost | An estimate of the cost of this project. This is a rollup calculation that sums estimated cost values for all tasks in this project. If no tasks are in the project, this field is editable. If tasks are configured, this field becomes a read-only rollup calculation and overwrites any earlier entry that you made. |
| Actual cost | The actual cost of this project. This is a rollup calculation that sums actual cost values for all tasks in this project. It remains editable on the project form at all times, regardless of project state or whether tasks are configured in the projects. |
| Budget cost | Budgeted cost for this project. This field is for project management planning only, serving as an initial value. It its not involved in any estimated cost or actual cost calculations for the project. Nor is this field used by Cost Management. |
| Description | A detailed description of the project. |
| Watch list | Allows users to subscribe to project notifications. |
| Work notes list | The list of users to receive email notifications when the work notes on the project are updated. |
| Work notes | Information about the milestones, impediments, or changes as the project progresses. |
| Activity | Tracks information or items not saved with a field in the record. |

| **Related Lists** | **Description** |
|---|---|
| Project Tasks | The tasks in the current project. Only the next-level tasks (immediate subtasks) appear in this related list. |
| Sub Projects | The child project records of the current project (available with the v3 application). |
| Baselines | A collection of all planned dates for all tasks and milestones at the time you create the baseline. |

| | |
|---|---|
| Stories | The list of stories in the current project. |

## Project Form Fields for Versions Prior to Fuji

The following table describes the project form fields fields from all views.

| Field (Default View) | Description |
|---|---|
| Number | A system generated with a configurable prefix. See Managing Record Numbering for more information about customizing number prefixes. |
| Company | [Required] Select the company this project belongs to. This field is in the v3 application (available starting with the Dublin release). |
| Time constraint | The field that indicates when a project is scheduled to start (available in the v2 application).<br>Select a starting time constraint from the following choices:<br>• **Start ASAP:** The project is scheduled to start immediately. When you create the project, the Project form uses **Start ASAP** by default and sets the **Planned start date** to the current date.<br>• **Start on specific date:** The project is scheduled to start on a date that you specify.<br>**Note:** The time constraint is for planning purposes. The project actually starts when you click **Start project** on the Project form.<br>This field is not available on the Project form starting with the v3 application. |
| Planned start date | The intended date the project should begin. This value is set to the earliest time that the project schedule allows (v3 application). For example, if the project task is created at 3 P.M. and the default schedule is in use (which has an 8 A.M. start date), the default task start is 8 A.M. the next day. ***Note:*** The planned start date must be within 15 years of the current date.<br>Click the calendar icon and select a date to start this project. Projects do not automatically start on the planned start date. The project actually starts when you click **Start project** on the Project form. |
| Planned end date | The intended date the project should end. ***Note:*** The planned end date must be within 15 years of the current date.<br>• **v3 application:** This field is editable but is automatically updated when the duration changes or when the planned start date changes (the planned duration is used to calculate the new end date).<br>• **v2 application:** [Read-only] The intended date the project should finish. The system calculates this field from the **Planned start date**, the **Planned duration**, and the **Schedule** (if selected). |
| Planned duration | The expected duration of this project. ***Note:*** The number of days cannot exceed 1500.<br>• **v3 application:** The duration is recalculated if the planned end date changes. The duration also considers the project schedule, accounting for any non-work time in the schedule. For example, if the default schedule is used, with a standard 8-hour work day, a project that starts at 8 A.M. on July 1 and ends at noon on July 2 is calculated as 1 day and 4 hours, not 28 hours. Any project or project task with no children is restricted to a maximum duration of 1500 days.<br>• **v2 application:** Enter the value in days, hours, minutes, or seconds. If tasks are present, this field is read-only, and the duration is calculated from the planned duration of the child tasks. The estimated duration shows total project time and takes the project schedule into consideration. |
| Phase | The project management phase the project is in. Select from the following: **Initiating**, **Planning**, **Executing**, **Monitoring/Controlling**, or **Closing**. The phase *does not* change automatically when you change the state. Change it manually. For example, a project could be in a closed state but still in the planning phase if you do not keep it up to date. |
| State | The current state of the project. All new projects begin as **Pending** and automatically change to **Work in progress** when you click the **Start project** button on the Project form.<br>Following are the default states: **Pending**, **Open**, **Work in Progress**, **Closed Complete**, **Closed Incomplete**, **Closed Skipped**. |
| Configuration item | The configuration item (CI) affected by this project. |
| Short description | A brief description of the project. |
| Description | A detailed description of the project. |
| Business case | The reason the project is underway. If there is a pressing need for this project from a business perspective, make the case here. |

| | |
|---|---|
| Work notes | Information about the milestones, impediments, or changes as the project progresses. |
| Planned effort | An estimate of how much time this project will take to complete. This is a rollup calculation that sums planned effort values for all tasks in this project. If no tasks are in the project, this field is editable. If tasks are configured, this field becomes a read-only rollup calculation and overwrites any earlier entry that you made. |
| Actual effort | The actual number of hours charged to the resources on this project. If you are using the Time Cards application, ServiceNow automatically calculates the value for this field using the totals for the time worked from the time cards of all the resources who worked on this project. |
| Budget cost | Budgeted cost for this project. This field is for project management planning only, serving as an initial value. It its not involved in any estimated cost or actual cost calculations for the project. Nor is this field used by Cost Management. |
| Estimated cost | An estimate of the cost of this project. This is a rollup calculation that sums estimated cost values for all tasks in this project. If no tasks are in the project, this field is editable. If tasks are configured, this field becomes a read-only rollup calculation and overwrites any earlier entry that you made. |
| Net value | The project's value to the company. This is expressed in expected revenue. |
| ROI% | [Read-Only] The system calculates the return on investment using the (net value/estimated cost) x 100 formula. To disable this feature, deselect the Calculate ROI percentage property. |
| Priority | A priority for this project: **Critical**, **High**, **Moderate**, **Low**, or **Planning**. The priority is for planning purposes only and is not determined by the priorities of the project tasks. |
| Actual start date | The date that this project actually began. If the project has not been started, this field is modifiable. At the time the project starts, this field becomes read only. |
| Actual end date | The date that this project actually ended. If the project has not been started, this field is modifiable. At the time the project completes, this field becomes read only. |
| Actual duration | The actual duration of the project from project start to project closure. As with planned duration, the actual duration shows total project time and takes the project schedule into consideration. |
| Actual cost | The actual cost of this project. This is a rollup calculation that sums actual cost values for all tasks in this project. It remains editable on the project form at all times, regardless of project state or whether tasks are configured in the projects. |
| Risk cost | The potential cost of the risk for this project. This field is for project management planning only, serving as an initial value. It its not involved in any estimated cost or actual cost calculations for the project, nor is this field used by Cost Management. |
| Schedule | • **v3 application:** the default schedule is an 8-hour work day (from 8 to 12 and 1 to 5). A day is considered as a working day, not a 24-hour day.<br>• **v2 application:** the optional schedule shows when project team members work on the project. Select a predefined schedule or create a new one on the Schedules table. The system uses the schedule to calculate the **Planned end date**. If no schedule is selected, the system calculates a planned duration of one day as 24 work hours. |

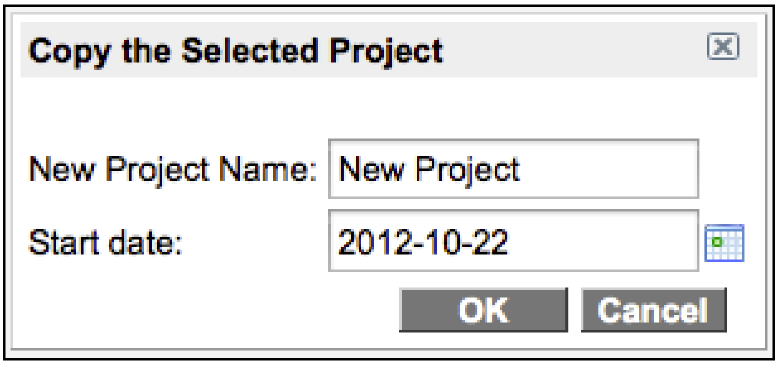
| Related Lists | Description |
|---|---|
| Project Tasks | The tasks in the current project. Only the next-level tasks (immediate subtasks) appear in this related list. |
| Sub Projects | The child project records of the current project (available with the v3 application). |
| Resource Plan | The resource plans for this project (available with the v3 application if resource management is active). |
| Group Resources | The group resources assigned to all the tasks in the project. When you add a resource to the **Assignment group** field on a Project Task form, the group appears in this related list (only in the v2 application). In the v3 application, this related list is not available. |
| User Resources | The user resources assigned to all the tasks in the current project. When you add a resource to the **Assigned to** field on a Project Task form, the user appears in this related list (only in the v2 application). In the v3 application, this related list is not available. |
| Baselines | A collection of all planned dates for all tasks and milestones at the time you create the baseline. |
| Goals | An optional related list of records that summarize one or more goals the project should accomplish. Goals are saved in the Goal table. |
| Risks | An optional related list of records for tracking any risks, such as cost overruns. Risks are saved in the Risk table. |
| Issues | An optional related list of records for issues related to the project. Assign a user resource and a priority level to each issue and track its progress with the **State** field. Issues do not appear on the Gantt chart, or project resource timeline, and are not used in any rollup calculations. |

| | |
|---|---|
| WBS List | The list of all project tasks and subtasks in their hierarchy (available with the v3 application). |

## Copying a Project

Another option for creating a project is to copy an existing project with all of its tasks and relationships. After you specify the start date for the copy, the system adjusts all task start and end dates automatically.

1. In the Project form, right-click the header bar and select **Copy Project**.
2. Enter a **New Project Name** for the new project that you are creating.
3. Select a **Start date**.
4. Click **OK**. The system creates the new project.



Copy the Selected Project

New Project Name: New Project

Start date: 2012-10-22

**Copy the Selected Project dialog box**

Copy partial project provides similar functionality. It copies all task or project relationships and children from the selected project and inserts them into the current project. In this case, a new project record is not created.

### Copied Fields

When you copy a project, all fields are copied over to the new project. Child tasks are defined with the same relationships, each lasting for the same duration as the original tasks. All project tasks are set to **Pending**. Task duration and the actual start and end dates are reset to null values. The state is set to **New** and percent complete is set to **0**.

Administrators can modify the **copy_project** UI page to determine which fields are reset or change the default values:

1. Navigate to **System UI > UI Pages**.
2. Open the **copy_project** record.
3. In the **Processing script** field, modify the values for `resetFields` or `defaultFields`. For example:

```
         /* resetFields is the array containing the list of names of
fields that need to be erased from the copied project tasks
         * defaultFields is the array containing the key, value pairs
of field names and values that need to be set on the copied tasks
         */
        var resetFields = new Array();
        var defaultFields = {};
        resetFields.push("work_start", "work_end", "work_duration");
        defaultFields["state"] = "-5";
        defaultFields["percent_complete"] = "0";
```

## Creating a Project from the Project Workbench

Navigate to **Project > Projects > Workbench** and click the **New Project** button in the project workbench header. For more information, see Creating a New Project from the Project Workbench.

## Creating a Project from the Demand Workbench

From the demand workbench, you can create a project from a qualified demand if you have the demand manager role. For more information, see Creating an Artifact from a Demand.

## Creating a Project from an Incident, Problem, or Change

If an incident, problem, or change is large enough to justify managing with a project, use the **Create Project** UI action on the incident, problem, or change form. All incidents, problems, and changes are part of the task table, which extends the [pm_project] table.

You can create a project from an incident, problem, or change by using

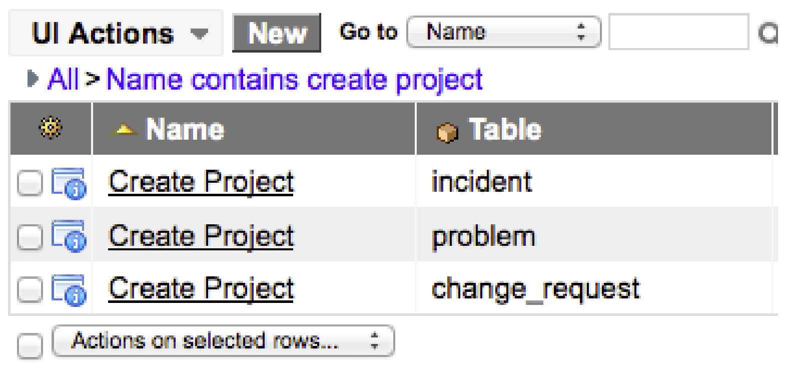
- The **Create Project** UI action
- The **Projects** related list

### Using the Create Project UI Action

This UI action is similar to the existing actions that allow users to create problem and change records from an existing incident record.

To configure the **Create Project** UI action:

1. Navigate to **System Definition > UI Actions**.
2. Open the **Create Project** UI action for the appropriate table (incident, problem, or change_request).


UI Actions
New
Go to
Name
All > Name contains create project
Actions on selected rows...

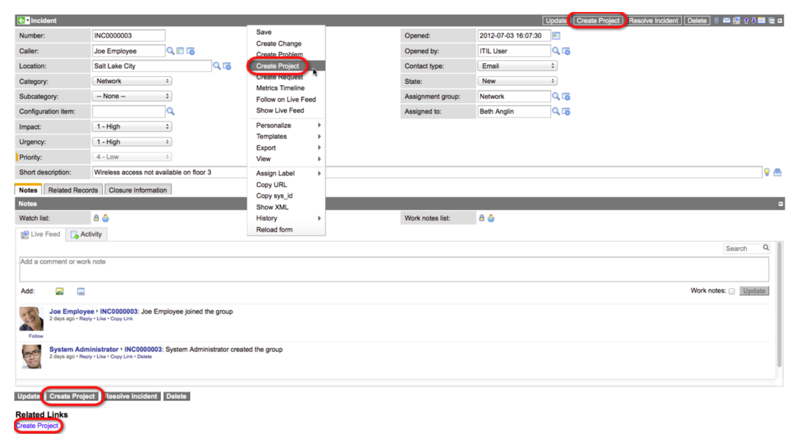
| Name | Table |
|---|---|
| Create Project | incident |
| Create Project | problem |
| Create Project | change_request |

**Create Project UI actions**

3. On the UI Action form, select the **Active** check box.
4. Select any of the following locations for the UI action:
   - **Form Context menu:** the action appears on the menu when a user right-clicks in the header bar of a record.
   - **Form button:** the action appears as a **Create Project** button on the form.
   - **Form link:** the action appears under **Related Links** on the form.

When a user clicks one of these links, the Project form appears with preconfigured data from the source table.

## Using the Projects Related List



Number: INC0000003
Caller: Joe Employee
Location: Salt Lake City
Category: Network
Subcategory: -- None --
Configuration item:
Impact: 1 - High
Urgency: 1 - High
Priority: 4 - Low
Short description: Wireless access not available on floor 3
Opened: 2012-07-03 16:07:30
Opened by: ITIL User
Contact type: Email
State: New
Assignment group: Network
Assigned to: Beth Anglin
Watch list:
Work notes list:
Work notes: □

Joe Employee ▸ INC0000003: Joe Employee joined the group
2 days ago • Reply • Like • Copy Link

System Administrator ▸ INC0000003: System Administrator created the group
2 days ago • Reply • Like • Copy Link • Delete

**Create Project UI actions on the incident form**

1. Navigate to the Incident, Problem, or Change Request form.
2. Configure the form to add **Project -> Parent** if the related list is not already present.
3. In the **Projects** related list, click **New**.
4. Fill in the Project form.

The project actually becomes a child task of the incident, problem, or change record.

**Note:** *Because a project view is automatically created when a project is created, the* ***New*** *button on the Projects related list is disabled (starting with the Fuji release).*

# What Do I Do Next?

## Add the Project Attachments Related List

Project participants can add attachments to project tasks and the project record itself. To view a list of all attachments in a project, configure the form and add the **Project Attachments** related list (available with the v3 application).

## Create Project Tasks

After the project is submitted, create the project tasks.

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/task/t_CreateAProject.html

# Creating Project Tasks

**Note:** *This article applies to Fuji. For more current information, see Project Tasks* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

## Overview

Tasks are the units of work that make up a project. The term *task* itself is arbitrary. For planning purposes, break down a project into several large tasks that can be referred to as phases or assignments. They, in turn, can be broken down further into smaller units of work. The size and number of tasks that comprise a project depends on the level of detail you want. For example, if part of a task requires a specific skill that is not required by the other activities in the task, it is a good idea to break down that task further.

To create and edit portfolios, projects, and tasks, users must have the project_manager role in their user profile record. See Installed with Project Management for more information on roles in the Project application.

### Key Concepts and Terms for This Topic

- **Project task creator:** a feature that allows you to create multiple tasks at the same time.
- **Project copy:** an option that allows you to replicate a project, including all tasks, relationships, and dependencies. Use this feature to quickly create a new project with tasks.
- **Assignment groups:** a collection of users with similar skills who could be assigned to the same types of project tasks.
- **Skills:** competencies assigned to groups or individual users. Assign skills to an individual user or to all the members of an assignment group. This helps you determine which users or groups should be assigned to tasks.
- **Dependencies and Relationships:** tasks can have nested relationships (parent and child tasks) and can also be dependent upon the completion of other tasks. See Project Task Relationships and Dependencies for more information.
- **pm_project_task:** the table where project task records are saved. This table extends the [planned_task] table.

## Approach to Tasking

You should decide on what approach you want to take to build a project. Consider the following approaches.

### Bottom-up (tactical) Tasking

This is the recommended approach for the Project application. Take this approach when you know what individual tasks need to be accomplished and you are more flexible about overall project duration and estimated cost. Use this approach to see how much a project will cost and how long it will take if you include every possible task. Project management supports tactical tasking by using rollup calculations on several project fields, such as project duration, so that the project adjusts to the tasks it contains.

To take this approach:

1. Use the Skills application to set up assignment groups and users.
2. Create the project.
3. Create the lowest-level tasks first. Use the default time constraint of **Start ASAP** rather than **Start on specific date** so that tasks remain flexible as you build relationships.

4. Assign resources who have the right skills to each task.
5. Continue to build low level tasks. When a set of tasks fits together into a larger unit, create another task that will serve as the parent task and add the smaller tasks as child tasks. See Parent-Child Task Relationships. For these parent tasks, and for the project itself, values like **Planned End Date** roll up to the parent automatically. This way, child tasks determine the duration and estimated cost for parent tasks and for the project.

### Top-down (strategic) Tasking

Take this approach when you want to build a project with fixed or inflexible time and budgetary constraints and well-defined phases. Establish well-defined milestones and dependencies between tasks that you should take into consideration from the beginning. Gradually add smaller tasks to the project at a later time. The main idea here is to avoid including all possible tasks in a project and stay flexible with what tasks should be included.

To take this approach:

1. Create the project with a specific start date and duration.

    **Note:** If you first create a project and then create the first task with a start-on date that is later than the project's start-on date, the project shifts later to start on the task's start date. To get around this, after you create the project, create the first task (or a milestone) with an ASAP start date. This ASAP task will hold the project start date in place while you add other tasks and sub-tasks. If you delete that ASAP task and no other tasks are holding the project start date in place, the project start date shifts to the date of the earliest task.

1. Create the highest-level tasks first with specific values for duration and estimated cost.
2. Create the necessary dependencies between tasks.
3. Create child tasks, if necessary, and create the resources or skills that you need at the same time through related lists.

Project management still rolls up several values, such as task duration. Therefore, if you create a task that has a longer duration than the project, the project adjusts to cover the entire duration of the task, which might defeat the purpose of this approach. Values are not rolled down from parent tasks, nor are there any restrictions on creating child tasks that are longer than specified duration of the parent.

## Default Values for New Tasks

When you create a task, the Project application automatically populates the following task fields:

- **Number:** the numerical reference to the task starting with **PRJTASK** followed by a number that is incremented one digit from the previous task record in the [pm_project_task] table.
- **Time constraint:** set to **Start ASAP**, indicating that the task starts after its predecessor task finishes (plus any defined lag time). See Project Task Relationships and Dependencies for more information about time constraints.
- **Planned start date:** set to the same time as the parent task, if any, allows (starting with the v3 application). If the project has a schedule, the new task's planned start date defaults to the earliest time allowed on the upcoming work day. For example, if the project task is created at 3 P.M. on Tuesday and the default schedule, which has an 8 A.M. start date, is in use, the task starts at 8 A.M. on Wednesday.

    For the v2 application, set to the date and time that you create the task.

- **Planned end date:** set to one day after the start date.
    - **v3 application:** This field is editable but is automatically updated when the duration changes or when the planned start date is changes (the planned duration is used to calculate the new end date).
    - **v2 application:** This field is calculated from the planned duration.
- **Planned duration:** set to 1 day by default. The duration is recalculated if the planned end date is changed, starting with the v3 application.

- **State:** set to **Pending**.

**Note:** *Any project or project task with no children is restricted to a maximum duration of 1500 days.*

# Creating Tasks

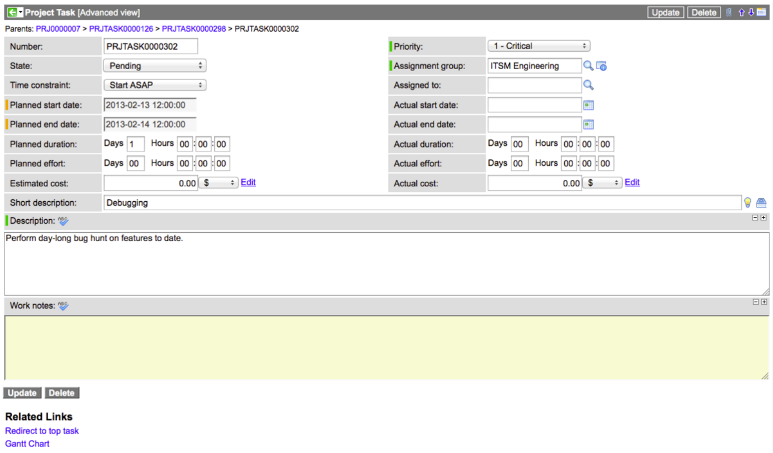
Project management provides the following ways for users with the project_manager role to create a task:

- Using the Project Tasks related list
- Creating a task from the Project Workbench
- Using the project task creator
- Copying an existing task
- Inserting a row into the Project Tasks list
- Using the Gantt chart
- Using the Project Tasks related list from an incident, problem or change
- Using a project task template

## Using the Project Tasks Related List

1. Navigate to **Project > Projects > All**
2. Select the project from the list.
3. In the **Project Tasks** related list, click **New**.
4. Enter a **Short description**.

   The short description identifies the task in records and in the Gantt Chart.
5. If necessary, change the **Priority** (for Eureka and prior releases).
6. To specify the skills required for this task, configure the form and add the **Skills** field.
7. Select a user resource for this task in the **Assigned to** field. The following conditions apply:
   - If an assignment group is defined, only users in that assignment group appear in the lookup list.
   - If skills are defined, only users with those skills appear in the lookup list. See Assigning Resources with the Right Skills for more information.
   - If no assignment groups or skills are defined, only users with one of the Project application user roles appear in the lookup list.
8. Enter a detailed **Description** of the task.
9. Click **Submit**.



**Project Task** [Advanced view]

Parents: PRJ0000007 > PRJTASK0000126 > PRJTASK0000298 > PRJTASK0000302

| | | | |
|---|---|---|---|
| Number: | PRJTASK0000302 | Priority: | 1 - Critical |
| State: | Pending | Assignment group: | ITSM Engineering |
| Time constraint: | Start ASAP | Assigned to: | |
| Planned start date: | 2013-02-13 12:00:00 | Actual start date: | |
| Planned end date: | 2013-02-14 12:00:00 | Actual end date: | |
| Planned duration: | Days 1 Hours 00 : 00 : 00 | Actual duration: | Days 00 Hours 00 : 00 : 00 |
| Planned effort: | Days 00 Hours 00 : 00 : 00 | Actual effort: | Days 00 Hours 00 : 00 : 00 |
| Estimated cost: | 0.00 $ Edit | Actual cost: | 0.00 $ Edit |
| Short description: | Debugging | | |

Description:

Perform day-long bug hunt on features to date.

Work notes:

**Related Links**

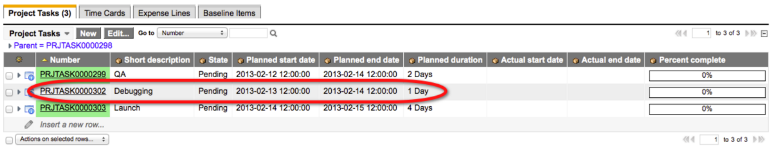
Redirect to top task

Gantt Chart

**A new project task**

| Field | Description |
|---|---|
| Short description | [Required] A brief description of the project task. |
| Number | A system generated with a configurable prefix. See Managing Record Numbering for more information about customizing number prefixes. |
| State | The current state of the project. The states include: **Pending**, **Open**, **Work in Progress**, **Closed Complete**, **Closed Incomplete**, **Closed Skipped**. |
| Planned start date | The estimated date and time for the project task to start. |
| Planned end date | The estimated date and time for the project task to end |
| Planned duration | The estimated length of time (from start time to end time) of the project task. |
| Percent complete | The percentage of the work that has been completed for the project task. |
| Assignment group | The group assigned to the project task. |
| Assigned to | The user assigned to the project task. |
| Actual start date | The date that the project task actually began. If the project task has not been started, this field is modifiable. At the time the project task starts, this field becomes read only. |
| Actual end date | The date that the project task actually ended. If the project task has not been started, this field is modifiable. At the time the project task completes, this field becomes read only. |
| Description | A detailed description of the project task. |
| Additional comments | Any additional information about the project task. |
| Work notes | Information about the project task as work progresses, such as milestones, changes, or impediments. |

The new task appears in the **Project Task** related list on the Project form. If time cards are in use, clicking **Submit** creates a time card for the resource.



Parent = PRJTASK0000298

| Number | Short description | State | Planned start date | Planned end date | Planned duration | Actual start date | Actual end date | Percent complete |
|---|---|---|---|---|---|---|---|---|
| PRJTASK0000299 | QA | Pending | 2013-02-12 12:00:00 | 2013-02-14 12:00:00 | 2 Days | | | 0% |
| PRJTASK0000302 | Debugging | Pending | 2013-02-13 12:00:00 | 2013-02-14 12:00:00 | 1 Day | | | 0% |
| PRJTASK0000303 | Launch | Pending | 2013-02-14 12:00:00 | 2013-02-15 12:00:00 | 4 Days | | | 0% |

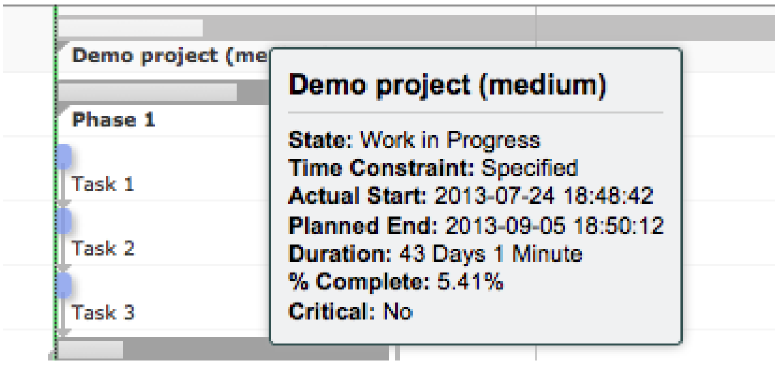
**The task in the Project Tasks related list**

## The Percentage Complete Field

As a good practice, add the **Percentage complete** field to the Project Task form if it is not already present on the form. This field lets you keep track of how much work has been completed on the task.

The value in this field is related to the **State** field in the v3 application (available starting with the Dublin release). If you change the percentage complete from **0** to any other value, the state of the task changes to **Work in Progress** when you save or update the record. Likewise, if you change the state of the task to **Closed Complete**, the value **100** is put in the **Percent complete** field.

This field is represented visually in the Gantt chart for parent tasks as a light-colored bar (the part of the task that is complete) over the darker, underlying bar (the full task). In this example, the Demo project is 5.41% complete as shown in the pop-up window, while the **Phase 1** task is about 50% complete (not shown in the pop-up window).



Demo project (medium)

**State:** Work in Progress
**Time Constraint:** Specified
**Actual Start:** 2013-07-24 18:48:42
**Planned End:** 2013-09-05 18:50:12
**Duration:** 43 Days 1 Minute
**% Complete:** 5.41%
**Critical:** No

A project showing the percentage complete values

### The Dependency Field

The Project Tasks list and the Project Tasks related list on the Project form include a **Dependency** field, starting with the Fuji release. Any dependencies for a task, such as a parent task, are displayed in this field. Clicking on a dependency takes you to that record. For more information, see Project Task Relationships and Dependencies.

## Creating a Project Task from the Project Workbench

If Project Portfolio Suite is activated, you can create a project task from the Project Workbench. For more information, see Creating a Project Task.

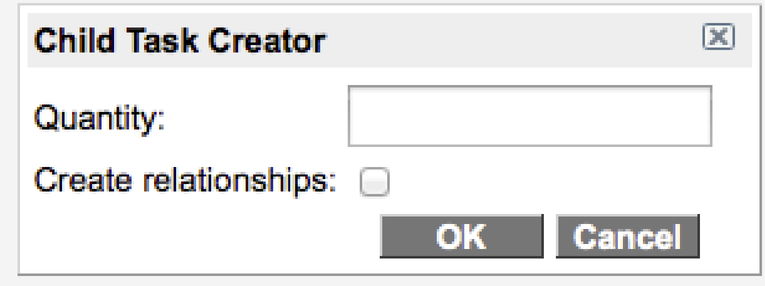
## Using the Project Task Creator

Use the project task creator to create multiple tasks at once.

1. In the Project form, right-click the header bar and select **Project task creator**.

   The Child Task Creator dialog box appears.
2. Enter the **Quantity** of tasks to create.
3. Select the **Create relationships** check box to create a *finish-to-start* dependency between these tasks (when the first task finishes, the next task starts). Clear the check box to create the tasks with no dependencies.
4. Click **OK**.

   The new tasks appear in the **Project Tasks** related list. ServiceNow automatically creates a task **Number** and a **Short description** that starts with **Auto created task**, followed by a number if more than one task is created.


Child Task Creator

Quantity:

Create relationships: □

**The project task creator**

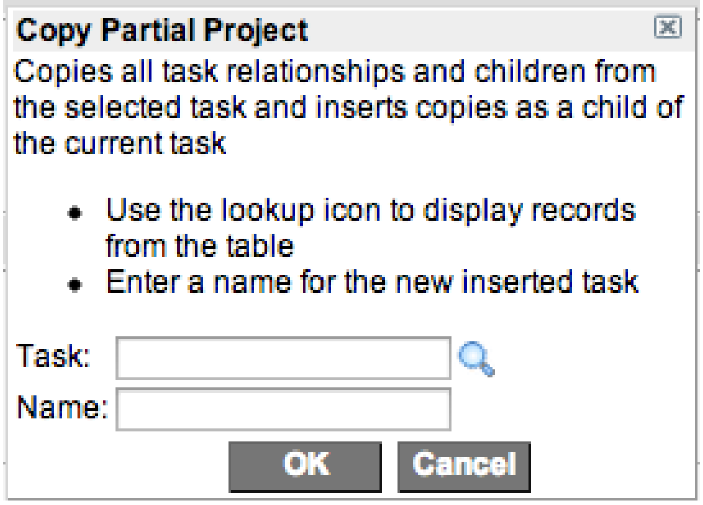
## Copying an Existing Task or Project

You can save time when building a project by copying tasks from other projects. The **Copy partial project** option copies a selected task, or even a whole project, if needed, including all child tasks. It also preserves all dependencies and relationships among the copied tasks and their child tasks. Any relationships or dependencies that involve tasks outside of the scope of the copied task are not preserved.

Copied tasks are inserted as a child of whatever task or project that you are currently viewing. Copied projects are added as a sub project of the current project or project task, which you can view in the **Sub Projects** related list.

1. In the Project or Project Task form, right-click the header bar and select **Copy partial project**.
2. In the **Task** field, select a project task or project to copy. By default, projects begin with **PRJ** and project tasks begin with **PRJTASK**.
3. Enter a **Name** for the new project or task.
4. Click **OK**.

- The copied task is added as a child of the current task or project.
- The copied project is added as a sub project of the current task or project.


Copy Partial Project

Copies all task relationships and children from the selected task and inserts copies as a child of the current task

- Use the lookup icon to display records from the table
- Enter a name for the new inserted task

Task:

Name:

**Copying a task and subtasks**

### Copied Fields

All fields are copied over to the new partial project. Task duration and the actual start and end dates are reset to null values. The state is set to **New** and percent complete is set to **0**.

Administrators can modify UI pages to determine which fields are reset or to change the default values:

1. Navigate to **System UI > UI Pages**.
2. Open the **copy_partial_project** record.
3. In the **Processing script** field, modify the values for `resetFields` or `defaultFields`. For example:

```
         /* resetFields is the array containing the list of names of
fields that need to be erased from the copied project tasks
         * defaultFields is the array containing the key, value pairs
of field names and values that need to be set on the copied tasks
         */
        var resetFields = new Array();
        var defaultFields = {};
        resetFields.push("work_start", "work_end", "work_duration");
        defaultFields["state"] = "-5";
        defaultFields["percent_complete"] = "0";
```

### Start Dates for Copied Projects

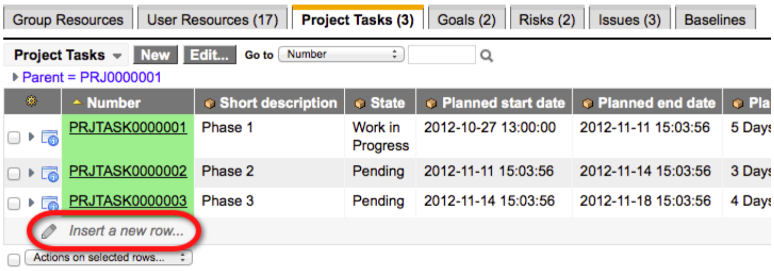
The start date for the set of tasks or whole project you just copied is set to the earliest possible start date:

- If the copied task has an ASAP start date, then the task starts as soon as the parent task, under which you imported the task, starts.
- If the copied task is scheduled to start earlier than the parent task, the copied task start is changed to the match the parent task start.
- If the copied task is scheduled to start any time after parent task starts, that specified date remains the same.

## Inserting a Row Into the Project Tasks List

Quickly create a new project task from the Project Task list. ServiceNow administrators must enable this feature. See Configuring List Control Settings for the List Editor for instructions.

1. In the Project form, navigate to the **Project Tasks** related list.
2. Double-click **Insert a new row**.
3. Click the green check mark.
4. Open the new task and edit the record as required.



Parent = PRJ0000001

| Number | Short description | State | Planned start date | Planned end date | Pla |
|---|---|---|---|---|---|
| PRJTASK0000001 | Phase 1 | Work in Progress | 2012-10-27 13:00:00 | 2012-11-11 15:03:56 | 5 Days |
| PRJTASK0000002 | Phase 2 | Pending | 2012-11-11 15:03:56 | 2012-11-14 15:03:56 | 3 Days |
| PRJTASK0000003 | Phase 3 | Pending | 2012-11-14 15:03:56 | 2012-11-18 15:03:56 | 4 Days |

**Inserting a new task into the Project Tasks related list**

## Using the Project Tasks Related List from an Incident, Problem, or Change

Just as you can create a project record from an incident, problem, or change, you can also create a project task.

1. Navigate to the Incident, Problem, or Change Request form.
2. Configure the form to add **Project Task -> Parent** if the related list is not already present.
3. In the **Project Tasks** related list, click **New**.
4. Fill in the Project Task form.

The project task actually becomes a child task of the incident, problem, or change record.

## Using a Project Task Template

You can save a project task as a template and reuse it when creating a new task.

1. If you do not have a template based on a project task, create one. Make sure it has many or all of the values for the fields you require.
2. Navigate to **Project > Tasks > All**.
3. Click **New**.
4. Right-click the form header and select **Templates > Apply Template > {template_name}**.

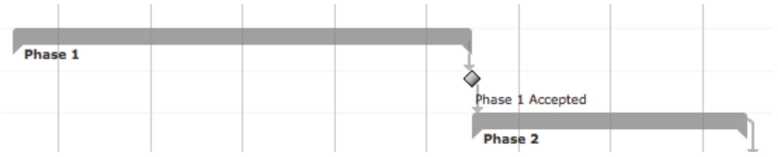
   The new task is created with the fields from the saved template, except the planned start and end dates.

   The task does not yet belong to a project. You must add it to the relevant project.

# Creating Milestones

## Gantt Chart

Milestones are project tasks with a duration of zero (0). Use milestones to mark key dates in your project, such as key decision points, approvals, and holidays. Milestones are treated like any other project task and you can create dependencies between tasks and milestones. On the Gantt chart, milestones are represented by a diamond.



**A dependency on a milestone**

**Note:** *Milestones cannot be shared between different projects unless one project is nested under another project.*

## Project Workbench

Milestones indicate important dates in a project and are represented by colored circles on the project workbench timeline. For more information, see Milestones.

# Assigning Resources with the Right Skills

**Note:** *The Resource Management application provides a more advanced way to plan for, request, and assign users and groups to tasks. The Resource Management application is available starting with the Dublin release.*

Assign configurable skills to an individual user or to all the members of an assignment group, and then assign a user who has the appropriate skills to complete the task. See Skills Management for instructions on creating skills and assigning them to users or groups. The instructions below explain how to select skills and assign users through the Project Task form. See Task Resources for more details on managing user and group resources and using the resource timeline.

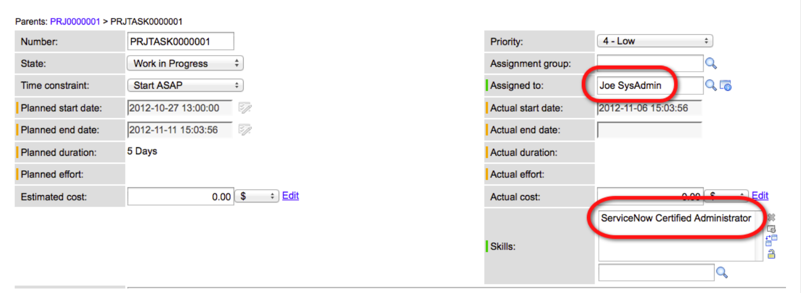
## Selecting Skills for a Task

To assign a skill to a task, a user or an assignment group must have that skill. The selection list for the **Assigned to** field on the Project Task form shows only the users and groups that have the skill selected. To select skills:

1. Open a project.
2. In the **Project Tasks** related list, open a task.
3. Configure the Project Task form and add the **Skills** field.
4. Click the padlock icon in the **Skills** field to enable selection.
5. Click the magnifier icon and select the skills required by this task.
6. Click **Update**.
7. Select a user in the **Assigned to** field.

   The picklist for this field is filtered for users who have the selected skills. This field has a reference qualifier that filters the picklist using the following logic:

- If an Assignment group is defined, then the list shows only members of that group.
- If Skills are defined, then the list shows only users with all the skills selected.
- If Assignment group and Skills are defined, then the list shows only group members who possess the defined skills.
- If the list does not include the correct user, verify the field qualifiers by editing the dictionary definition for **Assigned to** (right-click the field name and select **Configure Dictionary** (**Personalize Dictionary** in versions prior to Fuji). Also review the user record (**Skills > User**) to verify that the user has the correct attributes.



Parents: PRJ0000001 > PRJTASK0000001

| | | | |
|---|---|---|---|
| Number: | PRJTASK0000001 | Priority: | 4 - Low |
| State: | Work in Progress | Assignment group: | |
| Time constraint: | Start ASAP | Assigned to: | Joe SysAdmin |
| Planned start date: | 2012-10-27 13:00:00 | Actual start date: | 2012-11-06 15:03:56 |
| Planned end date: | 2012-11-11 15:03:56 | Actual end date: | |
| Planned duration: | 5 Days | Actual duration: | |
| Planned effort: | | Actual effort: | |
| Estimated cost: | 0.00 $ Edit | Actual cost: | 0.00 $ Edit |
| | | Skills: | ServiceNow Certified Administrator |

**Assigning a resource with a desired skill**

## Skills Inherited from an Assignment Group

It is not necessary to select a skill when an assignment group, with preconfigured skills, is assigned to the task.

1. Open a project.
2. In the **Project Tasks** related list, select a task.
3. in the **Assignment group** field, select a group.
4. In the **Assigned to** field, select a user to do the work.

   The selection list is filtered on the members of the assignment group. Any skills defined for the group are inherited by its members and are available to the task.

## Using a Schedule

Without an assigned schedule, a project calculates a day as a full 24 work hours. If you want to schedule tasks by a more realistic work day, assign a schedule to the project. If the schedules provided in the base system do not suit your needs, define a new one.

To add a schedule to the project:

1. Open a project.
2. If **Default** view is active, right-click the header bar and select **View > Advanced**.
3. Select a **Schedule**.

# Setting Up Project Task Notifications

It is good practice to send a notification to the user assigned to a project task when the task state changes, for example, from **Pending** to **Work in Progress**. You can activate default notifications or set up notifications with a workflow.

## Activating Email Notifications

The following email notifications for the Project application are available by default starting with the Dublin release, but are inactive. You must manually activate them.

| Notification | Table | Field | Condition | Description |
|---|---|---|---|---|
| Project task assigned | pm_project_task | Assigned to | Inserted or updated | Sends an email notification when a task is assigned to a resource or the assigned resource is changed. |
| Project task started | pm_project_task | State | Changes to **Work in Progress** | Sends an email notification when the project task starts. |
| Project task commented | pm_project_task | Additional comments | Any changes occur | Sends an email notification when the comment field is updated. |

## Setting Up Notifications with a Workflow Tool

Setting this up is simple with the workflow tool. This section provides an example of a workflow that sends an email notification when the state of a project task becomes **Work in Progress**. To configure a workflow with a notification:

1. Create a workflow with the following attributes:
   - **Name:** *Notify assignee*
   - **Table:** *Project task [pm_project_task]*
   - **If condition matches:** *Run if no other workflows matched yet*
   - **Condition:** *State is **Work in Progress** AND **Assigned to** is not empty*

   Do not modify other attributes in this example.
2. Add a single **Notification** activity between the **Start** and **End** activities. Drag the activity onto the connector line until it changes color.

   Its attributes are similar to the following:
   - **Name:** *Notify assignee*
   - **To:** *${assigned_to}.*
   - **Subject:** *Project task ${number} has been activated and is assigned to you.*
   - **Message:** *Project task ${number} has been activated and is assigned to you.*

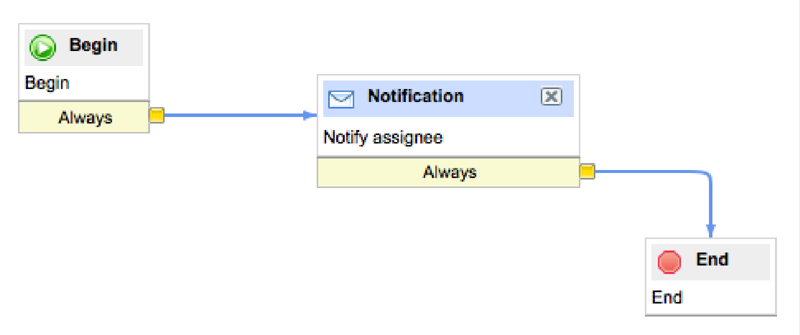
     Number: ${number}

     Short description: ${short_description}

     Planned start date: ${start_date}

     Planned end date: ${end_date}

     Planned duration: ${duration}



A notification workflow

## Linking Changes to a Project Task

Project managers can link change requests, created in the Change application, to project tasks in the v3 application, which is available starting with the Dublin release. Project tasks that link to change requests cannot also have child project tasks. Likewise, project tasks that already have child project tasks cannot also link to change requests. For more information, see Linking Changes to Project Tasks.

## What Do I Do Next?

After the project tasks are configured, create the necessary dependencies and relationships between tasks using the Gantt chart. You can also link changes to project tasks. For more information, see:

- Project Task Relationships and Dependencies
- Gantt Chart
- Linking Changes to Project Tasks

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_ProjectTasks.html

# Project Task Relationships and Dependencies

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Project Task Relationship and Dependencies*[1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

Project management enables you to create child tasks that are nested under a parent task and successor tasks that are dependent on the completion of a predecessor task. This page explains how to create such relationships and dependencies.

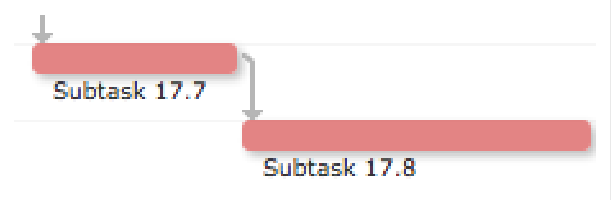
To create and edit portfolios, projects, and tasks, users must have the project_manager role in their user profile record. See Installed with Project Management for more information on roles in the Project application.

### Key Concepts and Terms for This Topic

- **Predecessor task:** a project task that, upon completion, is followed by another task. A predecessor task has a dependent relationship with its successor task.
- **Successor task:** a project task that cannot start until another task finishes. The successor task has a dependent relationship with its predecessor task.
- **Lag time:** a manually specified time break between predecessor and successor tasks. Configure a lag time, if necessary, when creating a predecessor-successor dependency.

  The system applies the project schedule in the v3 application (available starting with the Dublin release). For the v2 application, the system converts lag time to hours and does not consider the project schedule when applying lag time.
- **Parent task:** a project task with smaller tasks, referred to as child tasks, underneath it. Child tasks break down the work of a parent task into more manageable subsets. Certain fields for child tasks, such as planned end date, roll up and affect the same field in the parent task.
- **Child task:** a project task that is a subset of a larger task. Child task start dates cannot occur before the start date of the parent task.
- **Rollup task:** another term for a parent task in the context of aggregating child task items, such as effort or resources, into a larger parent task calculation. All fields on rollup task forms are read-only in the v3 application.
- **Roll down:** state changes roll down from the project to project tasks, and from parent tasks to child tasks in the v3 application.

# Task Dependencies



A finish-to-start dependency

A task dependency is created when one task is forced to start after another task finishes. For example, subtask 17.8 can only start when the **State** field of subtask 17.7 changes to **Closed**. The Project application only supports this kind of dependency, referred to as a finish-to-start dependency.

## Creating a Task Dependency

The easiest way to create a task dependency is with the Gantt chart.

Another option is to use related lists to create dependencies:

1. If the successor task does not already exist, navigate to the project form and create it.

   Do not create the task from the predecessor task form. Doing so creates a parent-child relationship.
2. Navigate to the predecessor task.
3. Configure the related lists for the Project Task form and add **Planned Task Relationship > Parent**. Do not select **Predecessor of** or **Successor of**.

   This adds the **Planned Task Relationships** related list to the Project Task form. This related list shows successor tasks.
4. In the **Planned Task Relationships** related list, click **New**.
5. On the Planned Task Relationship form, click the lookup icon and select the appropriate successor task.
6. Verify that the relationship **Type** is **Predecessor of::Successor**. *Do not* change this relationship type.
7. Enter the **Lag** time, if any, in either days or hours.
   - In the v3 application, the lag time takes the project schedule into consideration. If the lag time is 10 hours and the default schedule of an 8-hour work day is in use, the lag time pushes the task to the following day to cover the additional hours.
   - In the v2 application, the system converts the lag time to total hours. Therefore, a lag value of **1** day is equivalent to 24 hours. The lag time does not take the project schedule into consideration.
8. Click **Submit**.

## Task Time Constraints

The Project Task form includes a the **Time Constraint** field: either **Start ASAP** or **Start on specific date**.

- If the successor task is set to **Start ASAP**:

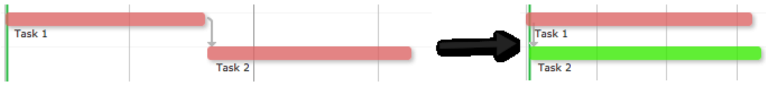
  The successor task appears on the Gantt chart as starting immediately after the predecessor completes without any lag time. However, the successor task can start on a later date if it has a value in the **Lag** field. To enter a lag value, double-click the relationship line in the Gantt chart and enter a **Lag** value. Alternatively, open the the predecessor task and enter a **Lag** value for the successor task in the **Planned Task Relationships** related list.
- If the successor task is set to a **Start on Specific Date** that is *later* than the finish date of the predecessor:

  The successor task starts at the time specified. On the Gantt chart, a lag appears just as if you set the **Lag** value on the relationship. However, the actual **Lag** value is not actually modified.
- If the successor task is set to a **Start on Specific Date** that is *earlier* than the finish date of the predecessor:

ServiceNow changes the successor task time constraint to **Start ASAP** and the task starts immediately after the predecessor finishes, unless a **Lag** value exists.

## State Changes on Tasks in Dependencies

Dependencies do not affect the ability to change the state of predecessor or successor tasks. For example, if a project is already in progress, you can still change a successor task to **Work in Progress** even if the predecessor task has not finished. Also modify the successor task to start on specified date that is earlier than the planned end date of the predecessor. Although this would violate the dependency for planning purposes, ServiceNow provides this kind of flexibility in modifying the project. You can also perform actions like closing a successor task, and then opening a predecessor task. Although you are allowed to make these kinds of modifications to predecessors and successors, the related project tasks and the way they are represented in the Gantt chart might show unexpected results.



A successor task prematurely changed to Work in Progress

## Modifying Dependencies

To modify an existing dependency from the Gantt chart:

1. Double click the relationship line in the Gantt chart.
2. Enter a different task in the **Successor** task field.

To modify an existing dependency from a related list:

1. Open the Project Task form for the successor task and click the existing predecessor task in the **Planned task relationships** related list.
2. Enter a different task in the **Successor** field.

## Removing Dependencies

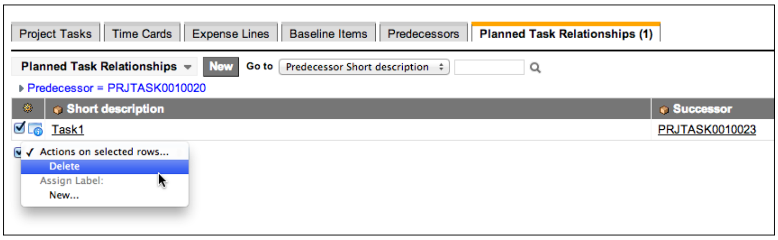
Remove a dependency that is no longer necessary from either the Gantt chart or the Project Task form. Removing the dependency also deletes the dependency record in the Planned Task Relationship table.

To remove a dependency from the Gantt chart:

1. Double-click the relationship.
2. In the **Planned Task Relationship** form, click **Delete**.

To remove a dependency from the Project Task form:

1. Open the predecessor task in the Project Task form and go to the **Planned Task Relationships** related list.
2. Select the check box beside the relationship being removed.
3. On the **Actions on selected rows** menu, select **Delete**.


Project Tasks
Time Cards
Expense Lines
Baseline Items
Predecessors
Planned Task Relationships (1)
Planned Task Relationships
New
Go to
Predecessor Short description
✓ Actions on selected rows...
Delete
Assign Label:
New...

Predecessor = PRJTASK0010020

| | Short description | Successor |
|---|---|---|
| ☑ | Task1 | PRJTASK0010023 |

Deleting the relationship record

## Parent-Child Task Relationships

If a task is relatively large and requires several users with different skills to manage, break the task into subtasks and create parent-child relationships. A child task should be a relatively smaller, manageable size of work. When you group child tasks together under a parent, values such as **Estimated cost** aggregate and roll up to the parent task. So the parent task takes on the form of a *summary task* or *rollup task* for its child tasks. **Planned start date** and **Planned end date** rollup occurs when you create child tasks: the duration of the parent automatically adjusts to *cover* its child tasks.

A parent-child relationship is different from a dependency relationship. In a dependency, one task must finish before another begins. In a parent-child relationship, any number of tasks can be nested under a parent task with or without any dependencies. When you create a parent-child relationship, the parent task number is saved in the **Parent** field in the Project Tasks table. All project management tasks have a parent: either another project task or the project itself.

## Creating a Parent-Child Task Relationship

The easiest method is to create parent-child relationships is on the Gantt chart.

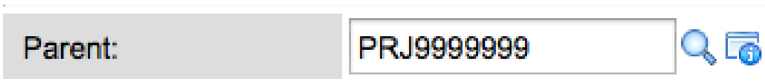
To create parent-child relationships with related lists:

1. Navigate to the parent task in the relationship.
2. In the **Project Tasks** related list, click **New**.

   The same Project Task form appears for all tasks regardless of the parent-child relationship.
3. Create the task and click **Submit**.

   The newly created task becomes the child task in the relationship.

To help remember what the parent of any task is, view the breadcrumb at the top of the Project Task form. It is also helpful to configure the form layout to include the **Parent** field.


Parent: PRJ9999999

**The Parent field. In this example the project is the parent.**

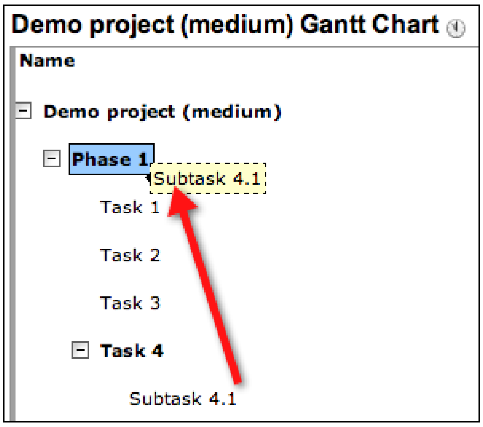
## Changing the Parent Task

To change the parent of a child task:

1. Navigate to the child task in the relationship.
2. Configure the form to add the **Parent** field, if needed.
3. In the **Parent** field, select the new parent task for this child task. To have the task stand alone in the project, select a project instead of a task.

## Modifying Parent-Child Relationships

To modify a parent-child relationship by using the Gantt chart, drag a child task to any other task (no matter whether it is a child or parent task). If you drag a child task up to the project name, it becomes a standalone task and is no longer considered a child task.



**Moving a child task**

To modify a parent-child relationship from a related list:

1. Navigate to the child task in the relationship.
2. Configure the form to add the **Parent** field if needed.
3. In the **Parent** field, select the new task that you want the child task assigned to. To have the task stand alone in the project, select a project instead of a task.

Unlike a dependency, a parent-child relationship is not saved as a record in any table. The only modification that takes place when a parent-child relationship is modified is the **Parent** field in the child task record.

## Time Constraints in Parent-Child Relationships

- If a child task is set to **Start ASAP**, the child task starts at the same time the parent task starts (as long as it does not have dependencies with other child tasks).
- If the parent task is set to **Start ASAP** and child tasks are set to **Start on specific date**, the earliest child task start date determines the start date of the parent (assuming no other dependencies). In this case, the parent's **Time constraint** field remains **Start ASAP** but the actual start date is changed to the start date of the earliest child task.
- If both the parent and first child task are set to **Start on specific date** but the first child starts later than the parent, the parent start date remains **Start on Specific Date** but the actual start date is pushed to the start date of the child. For example, if the parent task starts on October 1 and the earliest child task starts on October 2, the **Planned start date** of the parent is changed to October 2.
- Child precedence also applies to end dates. If the child task's estimated end date is later than the parent task's end date, the parent task's estimated end date extends to cover the child. For actual values, a parent has the same start date as the earliest start date of its children and the latest actual end date as the latest end date of its children, assuming the child tasks are **Closed Complete**. If the child tasks are not in the **Closed Complete** state, the actual end date of the parent is empty.
- For the planned start date of the parent task:
  - The planned start date is the earliest planned start date of all the children that do not have an actual start date.
  - If all child tasks have actual start dates, the parent task's planned start date is set to the actual start date.
- For the planned end date of the parent task, the latest planned end date or actual end date of the child tasks determines the parent's planned end date.

## Dependencies with Parent or Child Tasks

Options exist to create predecessor-successor relationships between child tasks with different parents, between two different parent tasks, or between a child task and another parent task. However, if the predecessor task finishes after the successor task starts, creating a dependency between child tasks that have different parents is not allowed.

# Parent-Child (Rollup) Task Calculations

Rollups involve date changes, state changes, and value calculations.

- Date changes involve modifying the planned start or end date of a parent task based on those values in child tasks.
- State changes involve modifying the state of the project record or parent task records if all child records are set to a certain state.
- Calculations involve summing the values of child tasks and then automatically updating the parent to reflect a new total.

Rollups work differently on these fields in the v3 application (available starting with the Dublin release):

- **Planned Start date:** set to read only for parent tasks. Remains editable for the project record (also considered the top-level task).
- **Planned End Date:** becomes read only.
- **Planned Duration:** becomes read only.
- **Actual Start Date:** becomes read only.
- **Actual end date:** becomes read only.
- **State:** becomes read only.

## Duration Rollups

Rollups are calculated for the following:

- **Planned duration and planned effort:** the sum of all planned duration and planned effort values for all child tasks.
- **Actual duration and actual effort:** the sum of all actual duration and actual effort values. Actual duration and actual effort values are calculated when all child tasks are in the **Closed Complete** state. Actual effort values can include rollups from time cards.

**Note:** *Verify that the time card property* ***Update the task's 'Actual effort' based on the hours entered in the time card*** *is enabled. Navigate to* ***Time cards > Administration > Properties*** *to enable this property.*

## Cost Rollups

Cost calculations roll up when the the Project Management Costing Add-on is active.

- **Estimated cost:** the sum of all cost estimates at the beginning of a project. Estimated costs of child tasks roll up to parent tasks and to the project.
- **Actual cost:** by default for the project, the sum of all costs of all the project's expense lines, which are typically associated with a a time card and a labor rate. To track costs, define rate cards for the task and labor expenses. These rate cards automatically generate expense lines showing actual expenditures, which are associated with the projects. If rate cards are defined, the task expense lines are generated as each project task closes, and labor expense lines are generated when time cards are approved. Expense lines are visible in the **Expense Lines** related list, which requires the **Advanced view** on a both Project and Project Task forms.

For actual costs of child tasks to properly roll up to the project and be added to project expense lines, the following must be true:

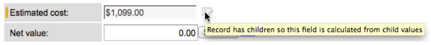
- The com.snc.project.rollup.cost property must be set to **true**. To enable this property, navigate to **Project > Administration > Properties** and select the **Enable project cost rollup** check box.
- The glide.cost_mgmt.process_task_top_task property must be set to false. To enable this property, navigate to **Financial Management > Admin > Properties** and select the **When creating a task expense line should the system also create expense lines for the task's top task** check box.
- The glide.cost_mgmt.calc_actual_cost property must be set to true. To enable this property, navigate to **Financial Management > Admin > Properties** and select the **For planned tasks types, calculate the actual cost field using the total of expense lines for the task** check box.

## Enabling Cost Rollup Calculations

To use rollup calculations:

1. Navigate to **Project > Administration > Properties**.
2. **Select Enable project cost rollup** and click **Save**.

Rollup values are read-only on forms. Point to the icon beside the field for a tooltip message.



Estimated cost: $1,099.00

Net value: 0.00

Record has children so this field is calculated from child values

**A rollup task for an estimated cost. The field is not directly editable on the parent.**

## Project State Rollups and Roll Downs

Project task states roll up. The state of parent tasks becomes read only in the v3 application, and changes automatically when you change the states of child tasks.

Project task states can roll up if:

- The state of the child task is manually changed and there are no other conditions on the parent task.
- The state of the child task is changed to **Work in Progress** or **Closed**. These states roll up to the parent. **Pending** and **Open** do not roll up to the parent task.

Project states can also roll down in the v3 application. If you change the state of a project to closed, all tasks under it change to the default closed value (**Closed Complete**). If a closed project or closed task is reopened, all tasks under it change as follows:

- Project or parent changed from closed to **Pending** or **Open**: child tasks change to **Open**.
- Project or parent changed from closed to **Work in Progress**:
  - Child tasks with a **Start on** date that has passed are changed to start **ASAP** and the state is changed to **Work in Progress**.
  - Child tasks with a **Start on** date that has not yet passed retain the same start on date but the state is changed to **Open**.

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_ProjectTaskRelationDepend.html

# Gantt Chart

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Gantt Chart* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

The Gantt chart is a visual representation of a project timeline that shows start and end dates of tasks, and the dependencies between tasks. Use the Gantt chart to add and delete tasks, change task dates and dependencies, and assess the progress of the overall project.

### Key Concepts and Terms for This Topic

- **Critical path:** linked project tasks that determine how long the project takes to complete.
- **Project portfolio Gantt chart:** a read-only Gantt chart available for a group of projects organized into a portfolio.

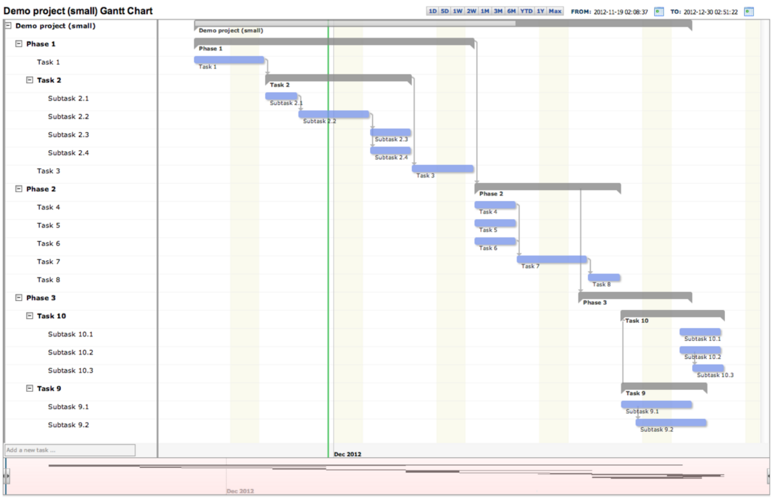
## Using the Gantt Chart

The Gantt chart is a visual representation of the project showing tasks, the amount of time to complete the tasks, and dependencies between the tasks.

### Navigating Through the Gantt Chart

1. Open a project.
2. Under **Related Links**, click **Gantt Chart**.

   The Gantt chart shows the project and tasks. Any nested projects (and associated tasks) are also shown within the hierarchical structure. Each task is labeled by its short description and is represented by a line proportional to the task's duration.
3. Change the perspective:
   - Move the slider at the bottom of the chart right or left to scroll across the chart, or adjust the end points of the slider to change the magnification. A narrow slider zooms in on the tasks and provides a more detailed view of the task relationships. A wide slider pulls the view out and makes more of the Gantt chart visible on the screen.
   - Click a time interval, such as **1M** for one month, from the options above the chart. The time interval zooms the chart in or out starting with the current time and date, which is signified by a vertical green line.
   - Specify a time interval by selecting **From** and **To** dates above the chart.



**The Gantt chart for a demo project that has not yet started**

On the Gantt chart, the following items are also available:

- The Project outline: a pane that lists all project tasks in a hierarchy.
- The project schedule: vertical light-brown bars in the background that show time periods that are not part of the work schedule. For example, the graphic above shows the Gantt chart for a project with the **8-5 weekdays excluding holidays** schedule specified. The vertical light-brown bars are weekends and holidays. Use this information when setting task start dates, end dates, and dependencies so that you do not schedule a task to begin or end on non-work days.

## Creating a Dependency in the Gantt Chart

To create a dependency between two tasks:

1. Click and hold a task.
2. Drag to a new task. A line appears showing the new relationship.
3. In the message that appears, confirm the new dependent relationship.
4. Continue connecting tasks for all dependencies in the project.

To edit a relationship:

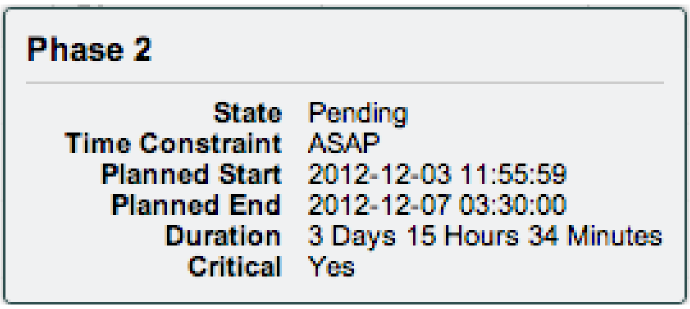
1. Double-click the connector between tasks.
2. Use the **Planned Task Relationship** dialog box to define the order of tasks and the lag time between tasks.
   - The relationship **Type** for planned tasks is **Predecessor of::Successor of** and should not be changed.
   - When a **Lag** value exists, ServiceNow recalculates the critical path accordingly.

## Viewing Information About a Task in the Gantt Chart

View full task information in either of these ways:

- Double-click a task bar to open the project task record and view or edit the form.
- For a brief summary of the task, point to a task bar to display a tooltip.

  Administrators can configure the information shown in the tooltip.



Phase 2

| | |
|---|---|
| State | Pending |
| Time Constraint | ASAP |
| Planned Start | 2012-12-03 11:55:59 |
| Planned End | 2012-12-07 03:30:00 |
| Duration | 3 Days 15 Hours 34 Minutes |
| Critical | Yes |

**Task summary**

## Adding Tasks

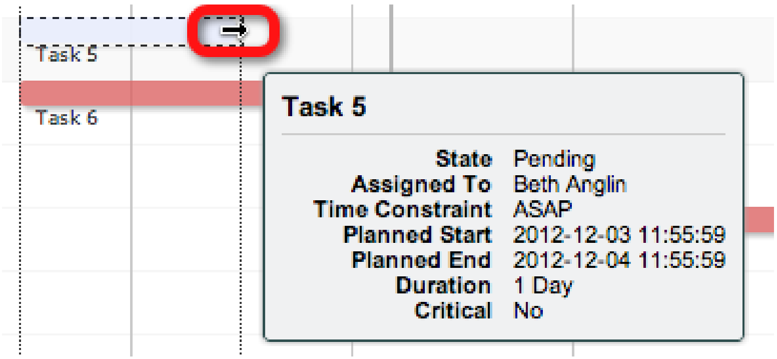
To add tasks to a project in the Gantt chart:

1. Navigate to the Gantt chart for the appropriate project.
2. Enter a name for the new task in the **Add a new task** field and press **Enter**.
3. From the left tab, drag the task name to the desired location in the task hierarchy.
4. Double-click the task bar and configure a duration in the Project Task form.
5. Click **Update**.
6. Create dependent relationships between tasks as appropriate.

## Editing Tasks

Use the Gantt chart to quickly change task attributes, such as start and end time, rather than opening every Task form and modifying field values one by one. From the Gantt chart, you can modify the following:

- **Planned start date:** move the task along the timeline to change the start time and to impose a **Time constraint** of **Start on a specific date**. You can also drag a task to change its start date if the task **Time constraint** is set to **Start on a specific date** (not **Start ASAP**) and the task has not yet started. The start date of a task cannot be modified if the task already started (has an actual start date), the task has already ended (has an actual end date), or the task time constraint is set to **Start ASAP**.
- **Planned end date:** drag the right edge of the task bar to extend the planned end date. You can only extend the planned of date for tasks that are not parent tasks and have not yet ended.



Task 5

| | |
|---|---|
| State | Pending |
| Assigned To | Beth Anglin |
| Time Constraint | ASAP |
| Planned Start | 2012-12-03 11:55:59 |
| Planned End | 2012-12-04 11:55:59 |
| Duration | 1 Day |
| Critical | No |

**Modifying the planned end date of a task**

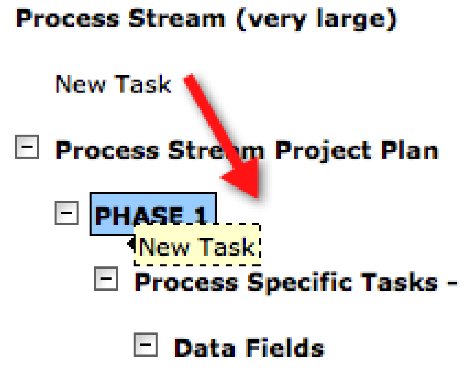
- **Dependencies**: To edit or delete a dependency, double-click an existing dependency connector line between two tasks and make the changes in the Planned Task Relationship dialog box. The relationship **Type** for planned tasks is **Predecessor of::Successor of** and should not be changed.
- **Lag time**: Lag time is the interval between the end of a predecessor task and the start of a successor task. The default setting is 0. To edit the lag time, double-click a connector and make the changes in the Planned Task Relationship dialog box.
- **Resources**: To change a user resource for an existing task or add a resource to a new task, double-click the task bar and edit the **Assigned to** field in the Project Task form.

## Editing Parent-Child Relationships

On the left pane, drag a task onto another task to change the parent-child relationship. The task that you dragged becomes the child task. Edit the task just like any other task.

Note the following:

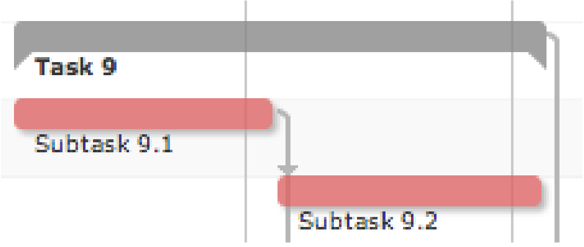
- New tasks have the **Time Constraint** field set to **Start ASAP**. If the task becomes a child task, the **Time Constraint** value stays **Start ASAP**, but the **Planned start date** becomes the start date of the parent task.
- You can further adjust the child task's start or end date with the dragging tool, but you can adjust the start date of the task only if the **Time Constraint** value is *not* **Start ASAP**.
- If a child task is scheduled to start on a date earlier than the parent task, the **Time Constraint** changes to **Start ASAP**.
- If you drag the end date of the child task past the end date of the parent task, the end date of the parent task automatically extends to the same end date.



**Dragging a new task under a parent task**

## The Critical Path

The critical path is highlighted in red on the Gantt chart to differentiate critical path tasks from standard tasks in blue and rollup tasks in gray. Not all tasks are part of the critical path, only those tasks that directly affect the finish date. Use the critical path to determine which tasks are driving the finish date. If schedule adjustments are necessary, consider making resource or other changes to those tasks on the critical path. In the following screenshot, the two child tasks are part of the critical path.



**The critical path appears in red**

## The Portfolio Gantt Chart

In addition to the project Gantt chart, the Project application offers a read-only project portfolio Gantt chart that shows the projects in each portfolio. One bar appears for each project in the portfolio.

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_GanttChart.html

# Work Breakdown Structure

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Work Breakdown Structure (WBS)* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

The work breakdown structure (WBS) is a hierarchical representation of all the tasks in your project. Use the WBS to get a quick overview of the entire project, including the tasks and subtasks nested in the project. The Gantt chart gives you a similar overview using graphical elements like lines and bars to show dependencies and lengths of tasks. The WBS is more data-driven, presenting a wide variety of task information in expandable rows. The WBS is included in project management v3, which is available starting with the Dublin release.

## Accessing the WBS

You can access the WBS list from the Project form. You can also view the WBS as a related list on both the Project and Project Task forms.

To view the WBS as a list:

1. Navigate to a Project form.
2. Click the **WBS List** related link.

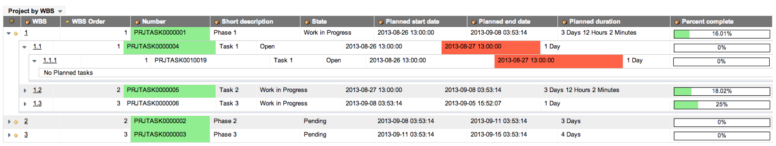
   The WBS list opens.

To view the WBS as a related list:

1. Navigate to a Project or Project Task form.
2. Right-click the header and select **View > WBS**.

   **WBS List** appears as a related list.

On the WBS, click the arrow icon ( ▶ ) to expand a task and view child tasks.



Project by WBS

| WBS | WBS Order | Number | Short description | State | Planned start date | Planned end date | Planned duration | Percent complete |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | PRJTASK0000001 | Phase 1 | Work in Progress | 2013-08-26 13:00:00 | 2013-09-08 03:53:14 | 3 Days 12 Hours 2 Minutes | 16.01% |
| 1.1 | 1 | PRJTASK0000004 | Task 1 | Open | 2013-08-26 13:00:00 | 2013-08-27 13:00:00 | 1 Day | 0% |
| 1.1.1 | 1 | PRJTASK0010019 | Task 1 | Open | 2013-08-26 13:00:00 | 2013-08-27 13:00:00 | 1 Day | 0% |
| No Planned tasks | | | | | | | | |
| 1.2 | 2 | PRJTASK0000005 | Task 2 | Work in Progress | 2013-08-27 13:00:00 | 2013-09-08 03:53:14 | 3 Days 12 Hours 2 Minutes | 18.02% |
| 1.3 | 3 | PRJTASK0000006 | Task 3 | Work in Progress | 2013-09-08 03:53:14 | 2013-09-05 15:52:07 | 1 Day | 25% |
| 2 | 2 | PRJTASK0000002 | Phase 2 | Pending | 2013-09-08 03:53:14 | 2013-09-11 03:53:14 | 3 Days | 0% |
| 3 | 3 | PRJTASK0000003 | Phase 3 | Pending | 2013-09-11 03:53:14 | 2013-09-15 03:53:14 | 4 Days | 0% |

The WBS for a demo project

Several important columns from the Project Task table appear by default. For example:

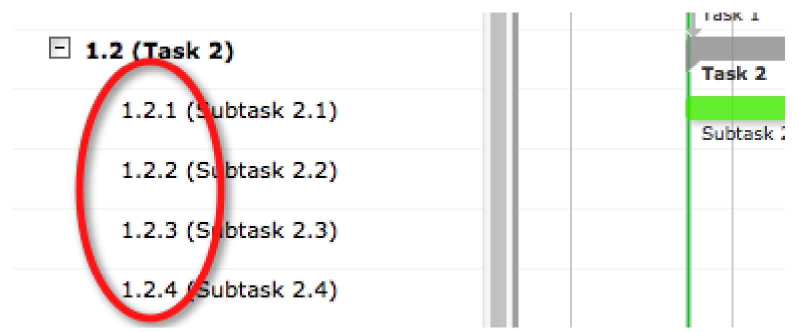
- **WBS:** The number of the task in the WBS hierarchy. The first task in the hierarchy is assigned the number **1**. Subtasks increment the number in the tenth place, such as **1.1** and **1.2**. The numbers are read-only.
- **WBS Order:** A number that represents the task in relation to its parent. The first subtask under a task has an WBS Order of **1**, and the next task **2**. If you edit a number, all tasks are moved accordingly after you refresh the list.
- **Number:** The task ID number, which should not be changed.

Other useful information also appears by default, such as the description and percentage complete. You can personalize the list like any other list by clicking the personalize list icon ( ⚙ ) and selecting the columns you want to view.

## Enabling WBS Numbering on the Gantt Chart

You can display the WBS number next to each task on the Gantt chart by enabling the following property:

1. Navigate to **Project > Administration > Properties**.
2. Select the **Display the Gantt chart in WBS ordering** check box.



WBS numbering on the Gantt chart

**Note:** *If you create a new task on the Gantt chart or change task order or relationships, the WBS number changes accordingly.*

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_WorkBreakdownStructure.html

# Task Resources

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Task Resources* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

*Resources* are the individuals assigned to perform tasks and subtasks in Project Management. You can manage your resources with resource plans in the Resource Management application, starting with the Dublin release.

In versions prior to the Dublin release, or if you are not using the Resource Management application, you can select resources from users or groups. **Skills** can be defined for individuals and for entire groups, ensuring that capable people are available for each project task. Because of the way in which Project Management is integrated into the task table, resources use basic ITIL task management processes. SLAs, approvals, and reporting are built in, just as they are for Incident Management and Change Management. A project task appears in a user's queue like any other task and does not indicate that it is a part of a project.

## Prerequisites

Before you perform the procedures in this page, be sure you have completed the **Gantt Chart** for your project, showing task relationships and durations.

## Adding User Resources

The User Resources record enables an administrator to associate a user with the project, with a project responsibility, and a percentage allocation. This percentage allocation checks against the project's schedule and calculates the amount of hours the percentage allocation represents. These hours are then used to determine whether this resource can continue to work project tasks, or if the resource is *out of time* for the project.

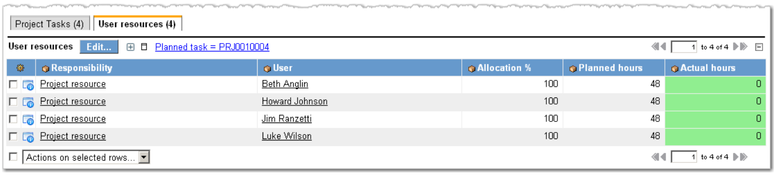
You add resources to a project in the **User resources** Related List in a project form. After selecting the project's resources, you attach those resources to their tasks in the **Resource Timeline**. Only one resource is assigned to a task at a time.

> **Note:** *You can add resources to* ***tasks*** *at the time the tasks are created. However, the best practice is to add multiple resources to the project through the Related List and then use the Timeline to make assignments, particularly for large projects. Resource allocation on large projects usually involves frequent adjustments.*

To create user resources from a project:

1. Click **Edit** in the **User resources** Related List.
2. In the slushbucket of users that appears, select the resources for the project and save the choices.

   The **User resource** list is populated with starting values of 100% allocation and zero hours.



Project Tasks (4) | User resources (4)

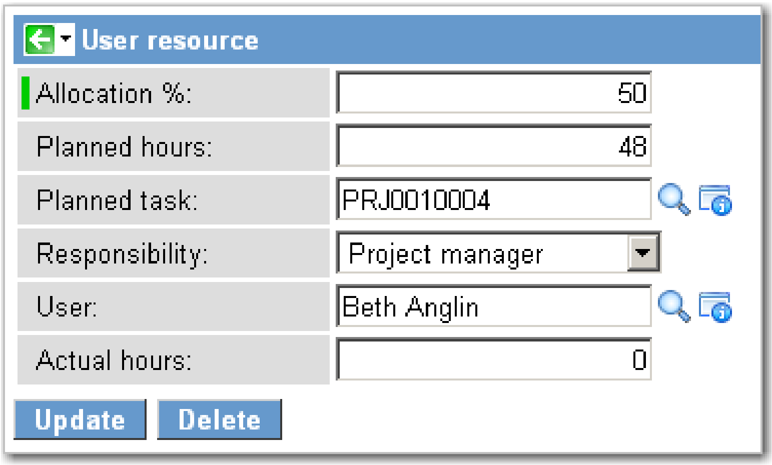
User resources Edit... Planned task = PRJ0010004 — 1 to 4 of 4

| | Responsibility | User | Allocation % | Planned hours | Actual hours |
|---|---|---|---|---|---|
| □ | Project resource | Beth Anglin | 100 | 48 | 0 |
| □ | Project resource | Howard Johnson | 100 | 48 | 0 |
| □ | Project resource | Jim Ranzetti | 100 | 48 | 0 |
| □ | Project resource | Luke Wilson | 100 | 48 | 0 |

□ Actions on selected rows... — 1 to 4 of 4

The User Resources related list

3. Click a link in the **Responsibility** column to open the User resource record.
4. Configure the resource's **Planned hours** and **Responsibility** (if different from **Project resource**).



User resource

| | |
|---|---|
| Allocation %: | 50 |
| Planned hours: | 48 |
| Planned task: | PRJ0010004 |
| Responsibility: | Project manager |
| User: | Beth Anglin |
| Actual hours: | 0 |

Configuring a user resource

5. Update the record and configure the next resource.

   The totals in the **Actual hours** column can be updated manually or automatically from the resource's **Time Card**.

## User Resources with Costing Add-on

The **Project Management v2 Costing Add-on plugin** adds an additional column called **Actual Cost** to the **User Resources** Related List. The platform calculates the **Actual Cost** of a resource by multiplying the **Actual Hours** consumed by the *labor rate* for the resource on the **Labor Rate Card** that applies to the user.

## Adding Group Resources

The **Group Resources** Related List in the Project form enables an administrator to associate ServiceNow groups with a project to facilitate resource planning. The following process illustrates how a large organization might use group resources to plan a project:

1. The project is created and the group resources are estimated, providing details for project review and approvals.
2. The project is approved, and the tasks are created without user resources being assigned. All that is necessary at this stage is to assign each task to a group that has the necessary skills.
3. Group managers review the work requirements and select the appropriate person to work on the task.

**Note:** *When a project task has a group in the* ***Assignment group*** *field, the list of users in the* ***Assigned to*** *field in the Project Task form is filtered on the members of the assigned group.*

Add group resources to projects at the task level and select the user resource to assign to the task at the same time. All group resources assigned to project tasks appear in the **Group Resources** Related List in the Project form.

1. Navigate to *Project* > *Projects* > *Pending* and select a Project.
2. In the Project record, select the **Project Tasks** Related List.
3. Select a task from the list.
4. Select a group from the **Assignment group** field.

   If the **Skills Management Plugin** is activated, an assignment group with pre-configured skills can be assigned to this task. For details, see **Assigning Skills to Tasks**.
5. Select a user from the **Assigned to** field.

   The selection list in this field is filtered on the members of the assignment group selected, ensuring that the user has the necessary skills required for the task.
6. Click **Update**.

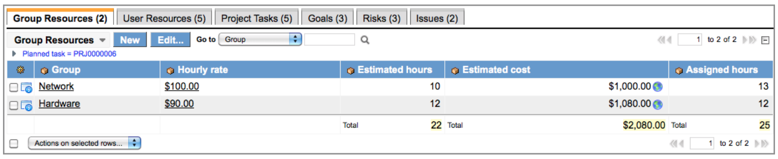
   In the Project form, the selected groups appear in the Group Resource Related List.

## Group Resource Related List

The fields provided in the **Group Resources** Related List depend on whether or not the Project Management v2 Costing Add-on plugin is installed.

| Field | Input Value |
|---|---|
| Group | Name of an assignment group defined for a task in this project. All members of an assignment group are available to work on a task and inherit any **skills** assigned to the group. |
| Hourly rate | Defines the hourly rate for members of this group, as defined in the Group record. This field requires the **Project Management v2 Costing Add-on plugin**. The hourly rate is multiplied by the **Estimated hours** for the group to estimate project costs. |
| Estimated hours | Estimated number of effort hours needed from this group to complete the project. This field is used during project planning to estimate resource requirements. The estimated hours are multiplied by the **Hourly rate** of the group to estimate project costs. This field is found on the **Group resource** form. Use one of the following methods to access this form:<br>• If the Project Management v2 Costing Add-on plugin is installed, click the link in the **Hourly rate** column.<br>• If the Project Management v2 Costing Add-on plugin is NOT installed, click the link in the **Estimated hours** column. |
| Estimated cost | Total of a group's **Hourly rate** multiplied by the **Estimated hours**. The total of each group's estimated cost is the estimated cost of the project. This field requires the **Project Management v2 Costing Add-on plugin**. For more information, see **Managing Project Costs**. |
| Assigned hours | The total of the hours assigned to each group from the **Planned effort** field in a Project Task form. This field is only visible in the Advanced View. Assigned hours represent specific time estimates on the task level that will differ from the **Estimated hours**. |

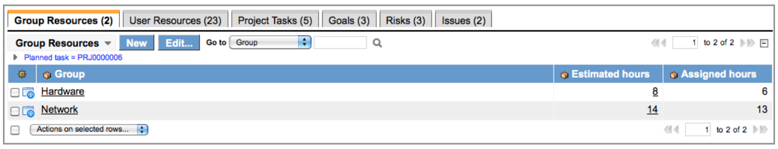
The following illustrates the Group Resources related list when the Costing Add-on is activated:



|  | Group | Hourly rate | Estimated hours | Estimated cost | Assigned hours |
|---|---|---|---|---|---|
| □ | Network | $100.00 | 10 | $1,000.00 | 13 |
| □ | Hardware | $90.00 | 12 | $1,080.00 | 12 |
|  |  |  | Total 22 | Total $2,080.00 | Total 25 |

Group Resources related list with Costing Add-on

The following illustrates the Group Resources related list without the Costing Add-on:



|  | Group | Estimated hours | Assigned hours |
|---|---|---|---|
| □ | Hardware | 8 | 6 |
| □ | Network | 14 | 13 |

Group Resources related list without the Costing Add-on

# Resource Timelines

The following updates were added to the User Resources timeline in the Spring 2010 Stable 2 release:

- Improved rendering performance, particularly for projects with a large number of items.
- Visual improvements including summary pane overlay bevel, timeline drop shadow, font adjustments, and preview mask coloring tweaks.
- Capability to split the timeline into multiple frames.
- Quick range selection buttons and start/end time view adjustment.
- Inner segments for Timeline Span elements on the timeline, useful for % complete on Gantt Charts or travel time for Field Service Management visual dispatching.
- Auto-refresh capability.

**Note:** *For more details and API documentation on the new features please see: Schedule Page Spring 2010 Stable 2 Updates*

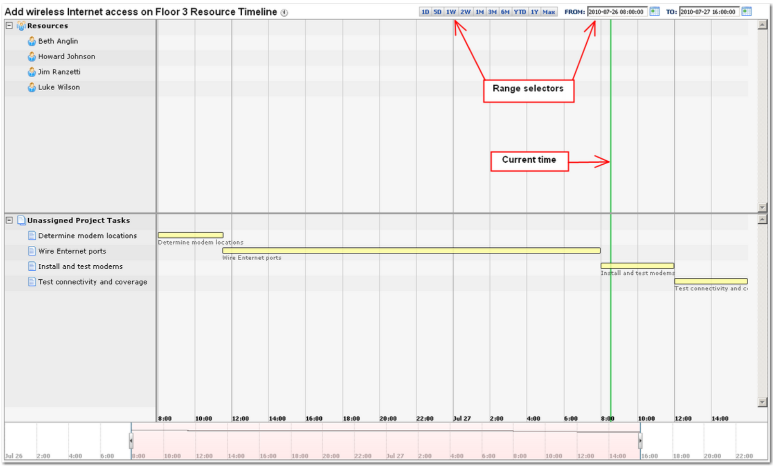
The *Resource Timeline* is a graphical, interactive map that establishes the relationships of tasks and sub-tasks to user resources for a project. Create the timeline after you identify the resources and tasks for the project. You must establish the relationships manually between the unassigned tasks and the free resources.

At this point, you should have a project record containing tasks and a list of resources to associate with those tasks.

1. Click the **Resource Timeline** Related Link in the project record.

   The Timeline displays two panes: *Resources* and *Unassigned Project Tasks*. You can resize the panes by dragging the dividing line that separates them. The tasks are arranged in the proper order (as defined in the **Gantt Chart**) and with the proper durations. In this example, the duration for the second task, **Wire Ethernet ports**, extends over two days, yet is configured for 8 hours. This is because we specified a project schedule of **8-5 weekdays**, and the time slot for that task must span the off-work hours between the days.

   **NOTE**: The **Schedule** field in the Project form is optional and visible only in the **Advanced** view of the form.



The resource timeline

2. Use the pink slider at the bottom of the timeline to change the perspective.

   a. Move the slider from right to left to view all the tasks on a long timeline.

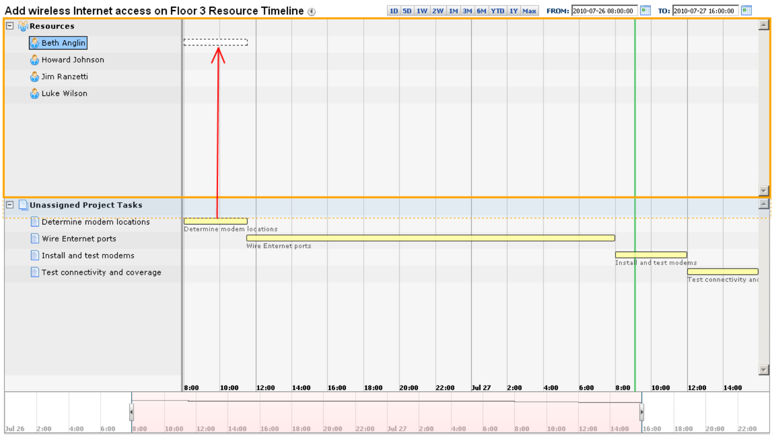
   b. Adjust the end points of the slider to change the magnification.

      A narrow slider zooms in on the tasks and provides a more detailed view of complex timelines. A wide slider pulls the view out and makes more of the timeline visible on the screen.

3. Use the Range Selectors at the top of the timeline to change the perspective.

   The increments go from one day to one year. To limit the timeline to the length of the current project, click **Max**. The green, vertical line is the current time and sweeps across the resource timeline automatically.

4. To assign a task to a resource, drag the task bar from the *Unassigned Project Tasks* pane up until the resource's name is highlighted, and then release the task bar.


Add wireless Internet access on Floor 3 Resource Timeline
1D 5D 1W 2W 1M 3M 6M YTD 1Y Max
FROM: 2010-07-26 08:00:00
TO: 2010-07-27 16:00:00
Resources
Beth Anglin
Howard Johnson
Jim Ranzetti
Luke Wilson
Unassigned Project Tasks
Determine modem locations
Wire Enternet ports
Install and test modems
Test connectivity and coverage

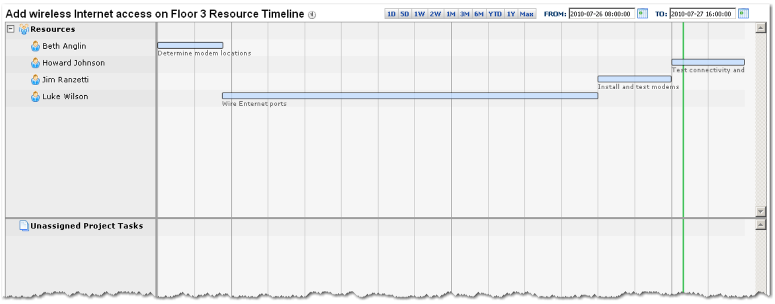
Assigning a task to a user

5. To *unassign* a task to a resource, grab the task bar with the mouse and move it back down into the *Unassigned Project Tasks* list. A dialog box appears asking you to confirm the action.

6. Click **OK**.

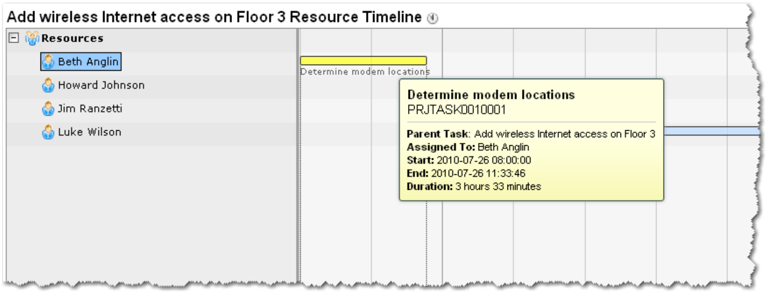
7. Assign the remaining tasks to resources.

   When you are done, the *Unassigned Project Tasks* pane is empty, and all the tasks are aligned with their resources.



Assigned project tasks on the timeline

8. To view the specifics of a task, hover the cursor over the task bar in the Timeline.


Add wireless Internet access on Floor 3 Resource Timeline
Resources
Beth Anglin
Howard Johnson
Jim Ranzetti
Luke Wilson
Determine modem locations

Determine modem locations
PRJTASK0010001

**Parent Task**: Add wireless Internet access on Floor 3
**Assigned To:** Beth Anglin
**Start:** 2010-07-26 08:00:00
**End:** 2010-07-26 11:33:46
**Duration:** 3 hours 33 minutes

Details of a project task

9. To edit or delete a task record, double-click on the bar.

   The task record shows a log of all resource assignments for that task in the **Activity** field.

## What Do I Do Next?

If you have finished selecting user resources for your project and have assigned tasks to the resources in the timeline, you are ready to **start the project**.

## Using Task Resources with Other Planned Tasks

If the Project Management v2 Plugin is activated, the Task Resources related lists and timeline are available on any planned task, using the method detailed above.

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_TaskResources.html

# Managing a Project

## Starting a Project

### Overview

Starting a project is the next step after setting up the project, populating it with tasks, and assigning resources.

### Starting a Project

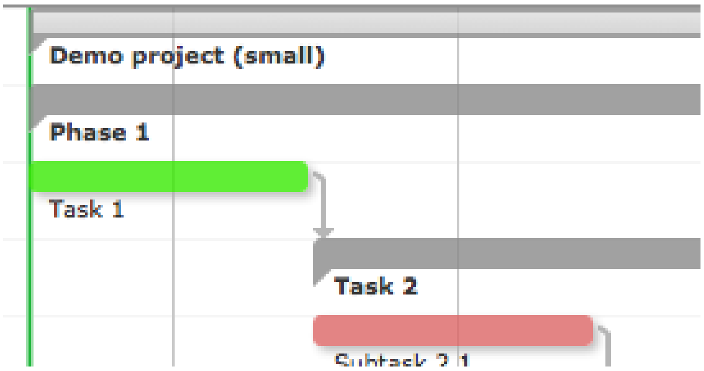
To start a project, navigate to the project form and do one of the following:

- Click the **Start Project** button
- Change the **State** field to **Work in Progress** and click **Update** to save the change.

### Changes That Occur After Starting a Project

- The read-only **Actual start date** field of the project changes to the current date.
- If a task or set of tasks are scheduled to start immediately upon project start (meaning that their time constraints are set to **Start ASAP** and they have no other start dependencies), the actual start dates of those tasks also change to the current date.
- The planned start dates of all other tasks adjust accordingly based on the time you started the project. Their new planned start dates depends on several factors, including dependent relationships with other tasks and the duration for each task.
- On the Gantt chart, the green line that represents the current date is lined up with the left edge of the bar that represents the project.

**Note:** *Once a project is in the* ***Work in Progress*** *state, it does not mean that the state of every task will start updating automatically based on their planned start dates. Other than the project tasks that you schedule to start ASAP when the project starts, project tasks are not started automatically. You must continue to manage the project and change each task's state to* ***Work in Progress****. Consider writing business rules to automate some of this functionality.*



**A project that was just started**

### Reopening or Restarting a Project

After a project is closed, you should not reopen it (changing the state to **Pending** or **Open**) or restart it (changing the state to **Work in Progress**). If you want to do so, a better practice would be to copy the project and manage it as a new project, adjusting the state of each task as needed.

If you do reopen or restart a project, note the following:

- **Changing the project to Work in Progress:** Project tasks with the **ASAP** time contraint that start when the project starts. Other tasks change to **Open**.
- **Changing the project to Pending or Open:** All tasks change to **Open**.
- **Changing a task to Work in Progress:** The project state changes to **Work in Progress**.
- **Changing a task to Pending or Open:** No changes occur to the project.
- **Actual values:** Actual start and end dates are reset. ASAP tasks begin as soon as they can and that date is reflected in the actual start date. Start on tasks remain in the open state.

# Project Portfolio Management

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Portfolio Management* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

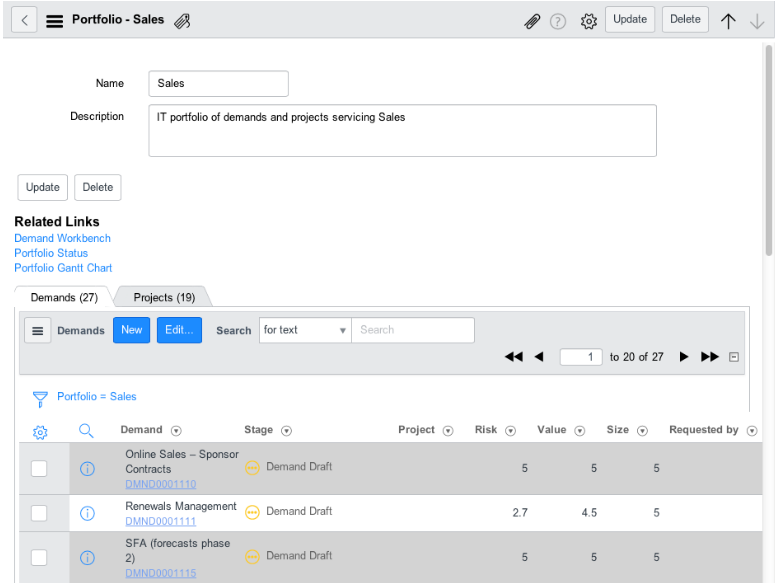
A portfolio is a collection of related projects. Portfolios also include demands starting with the Fuji release. Create a portfolio to measure the progress of several projects at the same time and to create multi-project reports for analysis. A read-only portfolio Gantt chart is available to view all projects in a portfolio.

## Creating a Portfolio

You can create a new portfolio from either the Project Management or the Demand Management applications.

### Creating a Portfolio in the Fuji Release

1. Navigate to one of the following:
   - **Demand > Settings > Portfolios**
   - **Project > Settings > Portfolios**
2. Click the **New** button.
3. Enter a unique **Name** that summarizes the projects and demands included in this portfolio.
4. Enter a **Description** for the portfolio that adequately explains the various projects and demands attached to it.
5. Click **Create Portfolio**.



**Portfolio - Sales**

Update | Delete

Name: Sales

Description: IT portfolio of demands and projects servicing Sales

Update | Delete

**Related Links**

Demand Workbench
Portfolio Status
Portfolio Gantt Chart

Demands (27) | Projects (19)

Demands | New | Edit... | Search | for text | Search

1 to 20 of 27

Portfolio = Sales

| Demand | Stage | Project | Risk | Value | Size | Requested by |
|---|---|---|---|---|---|---|
| Online Sales – Sponsor Contracts<br>DMND0001110 | Demand Draft | | 5 | 5 | 5 | |
| Renewals Management<br>DMND0001111 | Demand Draft | | 2.7 | 4.5 | 5 | |
| SFA (forecasts phase 2)<br>DMND0001115 | Demand Draft | | 5 | 5 | 5 | |

**Portfolio form**

| Field | Description |
|---|---|
| Name | The name of the portfolio. |
| Description | A detailed description of the portfolio that summarizes the included projects and demands. |
| **Related Links** | |
| Demand Workbench | Opens the demand workbench. |
| Portfolio Status | Displays the Portfolio Status list. These project views are typically used for reporting purposes to add the view to a portfolio dashboard. Project views are often associated with an actual project record and contain some of the same data, such as the planned start and end dates.<br>For Eureka and prior releases, you can create a project view that is not associated with any project and use the project view for reporting purposes only. You might want to do this if you are using another tool, such as Microsoft Project, to manage the project but you still want to include the project as part of a portfolio dashboard. Project views do not appear on the the project portfolio Gantt chart.<br>The **Portfolio Status** related link replaces the **Portfolio view** related list starting with the Fuji release.<br>To create a new project view, click this related link and then click **New** on the Portfolio Project form. |
| Portfolio Gantt Chart | Displays a Gantt chart with a summary of the portfolio's projects and milestones. |
| **Related Lists** | |
| Demands | Lists the demands included in the portfolio. Click the **New** button to create a new demand. Click the **Edit** button to add existing demands to the portfolio. |
| Projects | Lists the projects included in the portfolio. Click the **New** button to create a new project. Click the **Edit** button to add existing projects to the portfolio. |

## Creating a Portfolio in Eureka and Prior Releases

To create a new portfolio:

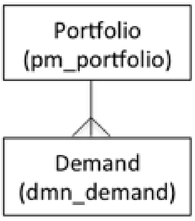
1. Navigate to **Project > Portfolios > Create New**.
2. Configure the following fields:
   - **Name:** Select a unique name that summarizes the projects included.
   - **Category:** Select one of the following categories:
     - **Transform business**
     - **Grow business**
     - **Run business**
   - **Active:** Select this check box to enable this portfolio. Only active portfolios appear in the list of available gauges for a homepage.
   - **State:** Select one of the following states: **Define**, **Analyze**, **Approve**, and **Charter**.
   - **Description:** Provide a detailed enough description of this portfolio to explain the various projects attached to it.
3. Click **Create Portfolio**.

# Projects and Demands in a Portfolio

Any project or demand can be added to a portfolio. The Portfolio form contains these related lists:

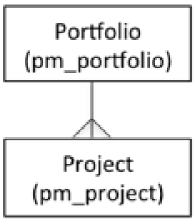
- **Demands:** (starting with the Fuji release) Demands that you create in the demand application with records in the Demand [dmn_demand] table. All demands in this related list are part of the portfolio.

  Demands are associated to a portfolio in a 1:m relationship, where one portfolio is associated to m demands.

- **Projects:** Projects that you create in the project application with records in the Project [pm_project] table. All projects in this related list are part of the portfolio.

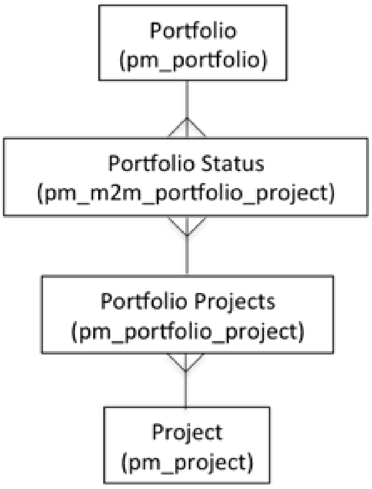
  Starting with the Fuji release, projects are associated to a portfolio in a 1:m relationship, where one portfolio is associated to m projects.

- **Project view:** (Eureka and prior releases) Views of projects that you create only from the Portfolio form. These project views are used for reporting purposes to add the view to a portfolio dashboard. Project views are often associated with an actual project record and contain some of the same data, such as the planned start and end dates. However, you can create a project view that is not associated with any project and use the project view. You might want to do this if you are using another tool, such as Microsoft Project, to manage the project but you still want to include the project as part of a portfolio dashboard. Project views do not appear on the the project portfolio Gantt chart. The **Project view** related list has been replaced by the **Project Status** related link starting with the Fuji release.

**Note:** *In the v2 application, a project view without an associated project is called an unattached project.*

A project record can be associated to multiple portfolio project records and any changes to the project propagate to the associated portfolio projects. Portfolio Status is a many-to-many relationship between the Portfolio Projects [pm_portfolio_project] and Portfolio [pm_portfolio] tables. Project is related to Portfolio through the Portfolio Projects and Portfolio Status tables as displayed in the following diagram.



The method of adding the project to a portfolio differs between the v2 and v3 applications. In the v2 application, you can add any existing project to the **Projects** related list. This action automatically creates associated project view. Starting with the v3 application, you must manually create a project view first and associate it with an existing project. After you save the project view, the project is added to the **Projects** related list and becomes part of the portfolio. The **Portfolio** field on the Project form is also populated with the portfolio name.

## Adding Existing Projects to a Portfolio

In the v2 application:

1. Navigate to the Portfolio form.
2. In the **Projects** related list, click **Edit**.
3. Select the projects from the **Collection** column and add them to the **Project List** column.
4. Click **Save**.

   The Project application creates a new record in the Portfolio Project table and adds this record to the **Portfolio View** related list.

In the v3 application in the Dublin release, the **Project** related list is read only. To add an existing project to a portfolio:

1. Navigate to the Portfolio form.
2. In the **Project View** related list, click **New**.
3. Fill out the form specifying the existing project in the **Project** field.
4. Click **Save**.

   Just as with the v2 application, the Project application creates a new record in the Portfolio Project table and adds this record to the **Portfolio View** related list. It also adds the project to the **Projects** related list, making that project part of the portfolio.

In the v3 application in the Eureka release, you must still add the project to the **Project View** related list in order to add the project to the portfolio itself. However, when you create a new project, the Project application automatically creates a new record in the Portfolio Project table as well. To add the project to a portfolio:

1. Navigate to the Portfolio form.
2. In the **Project View** related list, click **Edit**.
3. Select the project from the **Collection** column and add it to the **Project List** column.
4. Click **Save**.

## Adding Existing Demands to a Portfolio

1. Navigate to the Portfolio form.
2. In the **Demands** related list, click **Edit**.
3. Select the demand from the **Collection** column and add it to the **Demands List** column.
4. Click **Save**.

## Creating an Unattached Project in the v2 Application

To add a project to the **Project view** related list in the v2 application:

1. Navigate to the Portfolio form.
2. Under **Related Links**, click **Create Unattached Project**. The Portfolio Project form opens.

   The Portfolio Project form does not populate any data at the top of the form because it is not linked to a record in the Project table.
3. Configure the project details, such as planned start dates and effort, at the top of the form.

## Creating a Project View

1. Navigate to the Portfolio form.
2. Open the desired portfolio.
3. Click the **Portfolio Status** related link to open the Portfolio Status list.
4. Click **New**.
5. Fill in the fields on the Portfolio Project form, as appropriate (see table).
6. Click **Submit**.

| Field | Description |
|---|---|
| Project | The name of the project. |
| Short description | A brief description of the project. |
| Notes | Any additional notes pertaining to this project. |
| Planned start date | The intended date the project should begin. |
| Planned end date | The intended date the project should finish. |
| Risk | The level of risk for this project.<br>• Critical<br>• High<br>• Moderate<br>• Low<br>• Planning |
| Scope | Whether or not the project is still in scope. If the project scope has expanded and might impact project, select **yellow** or **red**.<br>• Green<br>• Yellow<br>• Red |

| | |
|---|---|
| Cost | The estimated cost of this project. If the project cost has grown and might impact the project, select **yellow** or **red**.<br>• Green<br>• Yellow<br>• Red |
| ROI % | The return on investment for this project. |
| % Complete | The percentage complete, which is copied from the Project record. |
| % Complete target | Where the project should currently be, based on its percentage complete. |
| Active | If the portfolio dashboard is active. |

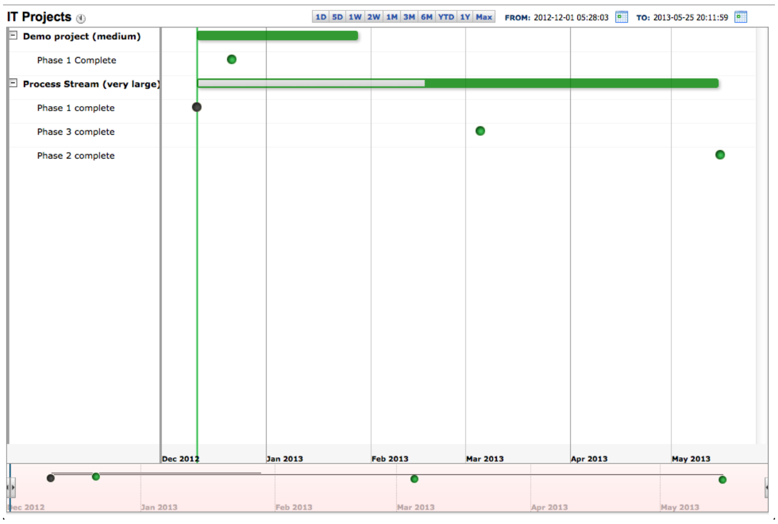
| Portfolio Dashboard Fields | Description |
|---|---|
| Update Dashboard | Option to update the fields in this section with the values from the top part of the form. Click this button to update the fields. |
| Risk | The level of risk for this project. |
| Risk comment | Any additional comments pertaining to the risk. |
| Scope | The scope status for this project. |
| Scope comment | Any additional comments pertaining to the scope. |
| Cost | The cost status for this project. |
| Cost comment | Any additional comments pertaining to the cost. |
| ROI % | The return on investment for this project. |
| ROI comment | Any additional comments pertaining to the return on investment. |
| % Complete | The percentage complete |
| % Complete comment | Any additional comments pertaining to the % complete. |

## The Project Portfolio Gantt Chart

Use the project portfolio Gantt chart to assess the progress of projects within a portfolio. The Gantt chart is similar to the project Gantt chart, but is read-only. The start and finish date for each project are shown in relationship to other projects. Each project is represented as a horizontal bar that is proportional to the project's duration. Project milestones appear as colored dots on the chart: completed milestones in black, missed milestones in red, and milestones that are on track in green.

1. Navigate to the Portfolio form.
2. Under **Related Links**, click **Portfolio Gantt Chart**.

   Alternatively, from the Portfolio List, right-click a portfolio name and select **Portfolio Gantt Chart**.



A portfolio gantt chart with demo projects and their milestones.

To synchronize the portfolio view with the data in the portfolio project records, click the **Refresh Portfolio** related link in the Portfolio record.

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_PortfolioManagement.html

# Project Portfolio Management Security

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Project Management* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

The Project Management application is a suite of tools that aids in managing projects, tasks, and resources. It provides the ability to create and manage projects of all sizes, from small projects with a few tasks to large portfolios of projects that contain complex tasks with various relationships and dependencies.

Project management includes tools to help you create, view, and manage projects:

- Project workbench gives project managers the ability to manage the different aspects of a project from a single page. This workbench supports both Project Management and Application Lifecycle Management applications, allowing for a hybrid approach to project management. Project managers can create projects that combine both Waterfall and Agile methodologies by using Waterfall, Agile, and Test phases. The project workbench is available starting with the Fuji release.
- Project templates define the basic structure of a project, including project tasks and sub-tasks, attachments, and other project information. The project template feature gives project managers a simple way to create, save, and reuse this project structure.

Project Management also includes features that enable you to achieve your project goals in alignment with the other activities your organization is managing with ServiceNow, such as:

- Integration with other features and applications on the ServiceNow platform, such as change management, resource management, and reports.
- Easy-to-read Gantt charts and Work Breakdown Structure lists that help you visualize large projects with complex relationships and dependencies.

## Video Tutorial

This video provides a brief tour of the Project application in the Eureka release.

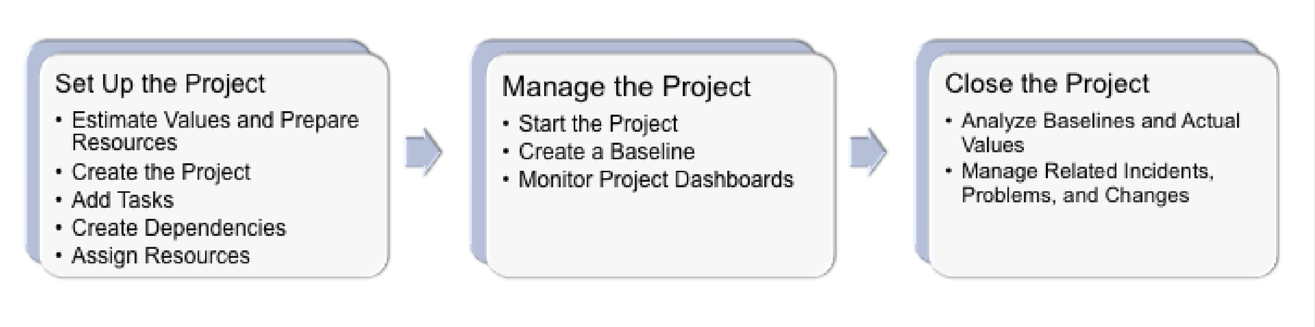
## Key Terms

- **Portfolio:** a collection of projects managed as a group to achieve strategic and operational objectives.
- **Project:** any planned, collaborative effort that is designed to achieve an objective.
- **Agile project:** any planned, collaborative effort that is designed to achieve an objective and uses Agile.
- **Project workbench:** a single page that presents project information in two panes. The the upper pane includes a project timeline and displays the project phases and milestones. The lower pane presents details about the currently selected phase in either a list view or visual task board.
- **Phase:** one stage or one segment of a project. Three types of phases can be added to the timeline in the project workbench:
  - **Waterfall phase:** a waterfall phase contains project tasks. A project can have multiple waterfall phases.
  - **Agile phase:** an agile phase contains stories. A project can have one Agile phase.
  - **Test phase:** a test phase contains test cases and can also include a team assignment. A project can have one test phase.
- **Story:** a brief statement of a product requirement or a customer business case that is used in the scrum method of agile software development. Typically, stories are expressed in plain language to help the reader understand what the software should accomplish.
- **Task:** a unit of work within a project. Projects typically contain several tasks.
- **Test case:** a collection of related tests. A test case is saved as part of a test suite and can be added to a test plan.

## Basics of Project Management

The Project application helps you plan and track projects, plus it integrates with other ServiceNow applications. For example, if an incident, problem, or change is large enough to require an entire project to manage, create projects from an incident, problem, or change form.

There are several paths available to manage a project. The best path usually depends on business needs. The steps below are designed to get a project up and running with the minimum amount of effort. Alternative methods to these procedures are also explained.


Set Up the Project
- Estimate Values and Prepare Resources
- Create the Project
- Add Tasks
- Create Dependencies
- Assign Resources

Manage the Project
- Start the Project
- Create a Baseline
- Monitor Project Dashboards

Close the Project
- Analyze Baselines and Actual Values
- Manage Related Incidents, Problems, and Changes

# Set Up the Project

Setting up a project involves deciding on an approach for creating and linking project tasks and making sure the necessary users and groups are created in ServiceNow so you can assign them to project tasks.

## Plan the Project

Before creating a project, consider the following questions and issues:

- **Do you want a top-down or bottom-up approach to tasking?**

  Top-down tasking involves creating a project first, then identifying major project phases. Later on, phases can be broken down into tasks and subtasks. The emphasis is on creating estimates for high-level items such as phases and parent tasks and then building the project down from there toward a more detailed level. Use caution when creating tasks for top-down tasking. If you first create a project and then create a task under it with a start-on date later than the project's start date, the project shifts later to start on the task start date. The Project application supports bottom-up tasking better.

  Bottom-up tasking involves creating several sets of small tasks and estimating task items such as effort, cost, and duration. These estimations are then aggregated into high-level parent tasks (rollup tasks) and phases. The emphasis is on estimating smaller chunks of work as accurately as possible first, then letting those estimations roll up into parent tasks, phases, and the project itself.

- **Is the project part of a larger portfolio of projects?**

  Also consider portfolio planning and how the project relates to similar projects or initiatives.

- **What types of dependencies will the tasks have with other tasks?**

  The Project application supports only finish-to-start dependencies.

- **Can milestones and project baselines help manage a project?**

  A milestone is a project task with a duration of 0. Use milestones to indicate important dates in a project. If necessary, create dependencies between tasks and milestones so that a task does not start until a milestone has been reached.

  A baseline is a snapshot of each task's current planned start and end dates at the time the baseline was created. An line appears under each task on the Gantt chart for the original planned start and end dates. The line appears shifted to the left or right depending on whether the task was started early or late. If tasks slip to later dates, the baseline indicator provides an easy way to see how severe the delays will be.

- **Have the necessary skills, groups, and resources been created in ServiceNow?**

  If project tasks will be assigned to different groups or individual resources with the required skills, create users and groups and configure the Skills Management application.

- **Does an existing incident, problem, or change justify creating a project in order to track it?**

  Of these record types, a change is most likely to lead to activities that should be tracked as a project.

- **Do you want to track project costs?**

  Estimate group resource costs before starting the project and then track the actual cost of each user resource from time cards. The Project application can also calculate the costs of affected CIs in a project. The Project Management Costing add-on is required to track costs.

- **What goals do you want the project to achieve?**

  Every project should have at least one goal. Project goals are saved in the Goal table and can link to any task. In a typical scenario, link one goal to each project and keep the goal's **State** field up to date.

### Create the Project

After choosing an approach and gathering initial estimates for the planned start date, estimated cost, and a well-defined business case, create the project in the Project application or in the project workbench.

### Add Project Tasks, Dependencies, and Relationships

After creating a project record, create tasks.

- For top-down planning, create a task that you already know will include several child tasks. Then create the child tasks and specify that they are child tasks of the first task you created.
- For bottom-up planning, create tasks for the smallest units of work first. Then you can create *intermediary* parent tasks that cover a group of related child tasks. For example, if there are five sequential tasks that comprise a phase of a project called *install database*, create the five tasks first. Then create another task called *Database installation* and make it the parent task of the five tasks. Rollup calculations, such as **Planned duration**, for the *Database installation* task are automatically calculated based on the child tasks.

It is easiest to build task relationships and dependencies while creating sets of tasks.

- A dependency means one task is forced to start after another task finishes. This is the only type of dependency ServiceNow supports.
- A relationship means a parent-child relationship whereby several subtasks are configured under a parent task or phase, which rolls up fields like **Planned duration** and **Estimated cost**.

Use the Gantt chart in conjunction with task forms and related lists to build relationships. Add milestones based on the project's major events and create dependencies between milestones and tasks, if necessary. See Project Task Relationships and Dependencies and Gantt Chart for more information.

Also set up notifications to alert project task assignees when their tasks move to the **Work in Progress** state. See Creating Project Tasks for more information on creating tasks.

### Assign Resources or Assignment Groups to the Tasks

User resources are the individuals in an organization who are assigned to project tasks. You can manage your resources with resource plans in the Resource Management application, starting with the Dublin release. In versions prior to the Dublin release, or if you are not using the Resource Management application, you can select resources from users or groups.

See Working with Resource Plans if you are using the Resource Management application. Otherwise, see Task Resources.

### Add the Project to a Portfolio

A portfolio is a group of related projects. If the project is related to other projects, create a portfolio and add the project. The Project application provides a useful portfolio view that makes its easy to report on the status of all projects in a portfolio. Portfolios also include demands starting with the Fuji release.

## Manage the Project

After the preceding steps are complete, the project can be started. To measure the project against initial estimates, create a baseline, which is a snapshot of the entire project including all planned dates for all project tasks and milestones. The project manager can manage a project from the project workbench starting with the Fuji release.

### Start the Project

Start the project by clicking **Start project** on the Project form or changing the project state to **Work in Progress**. Starting the project changes the **State** field on the Project form to **Work in Progress** and changes the **Actual start date** of the project to the current date. See Starting a Project for more information.

### Monitor the Project and Customize Dashboards

ServiceNow provides the ability to update important project status information, such as the number of milestones slipped. It also provides summaries for cost, scope, project risk, and so on. Modify this information as needed with the **Portfolio View** related list on the Portfolio form and display this information on the **Project Overview** homepage. In addition, use the project reports installed with the application, such as **Active projects** or **Projects (by priority)**, to show important project information.

When the project is underway, continue to access project records and make changes to several items, including costs, priority, schedule, and planned values that are not rollups. Keep detailed project records for risks and issues and refer back to them after a project is complete. Also create baselines along the way to easily see if any project phases or tasks are slipping at the time you create the baseline.

- See Project Reporting for more information on the available reporting options.
- See Project Portfolio Management for more information on viewing a summary of all project information in a portfolio, including completed and slipped milestones.
- See Updating a Project in Progress for more information on what is necessary while a project is underway.

## Close the Project

When the project is complete, change its state to **Closed complete** on the project form. When a project is in the closed state, the Project application calculates actual values like **Actual duration**.

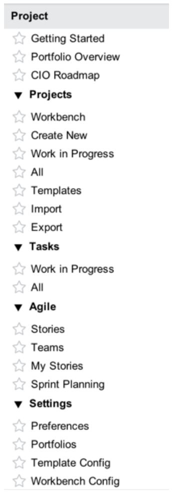
Post-project activities include analyzing project baselines and actual values and generating a final project dashboard. If the project was successful and can be used as a template for future projects, make a copy of it.

If the project was created from a change, incident, or problem record, there are several other activities you may need to perform in ServiceNow. See Closing a Project for more information.

# Menus and Modules

Activating this feature adds the Project Management menu to the application navigator with the following modules.

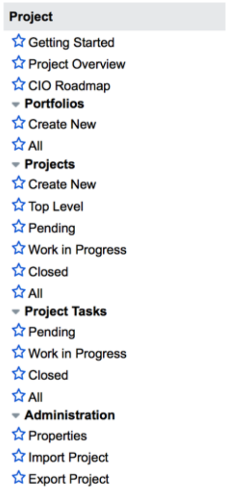
## Fuji Menu



The Project application (Fuji)

- **Getting Started:** Access the wiki documentation for the Project application.
- **Portfolio Overview:** Open the Project Overview homepage, which contains several built-in reports.
- **CIO Roadmap:** Open the CIO Roadmap.
- **Projects**
  - **Workbench:** Open the project workbench.
  - **Create New:** Create a new project.
  - **Work in Progress:** View projects currently in progress.
  - **All:** View all project records.
  - **Templates:** View project templates.
  - **Import:** Import a Microsoft Project.
  - **Export:** Export a ServiceNow project to be used in Microsoft Project.
- **Tasks**
  - **Work in Progress:** View project tasks currently in progress.
  - **All:** View all project task records.
- **Agile:** this module appears if the Project application is activated as part of the Project Portfolio Suite.
  - **Stories:** View all stories.
  - **Teams:** Display the Teams list, which shows a list of current teams.
  - **My Stories:** View a list of stories assigned to the current user.
  - **Sprint Planning:** Open the Sprint Planning page for the selected team.
- **Settings**
  - **Preferences:** Edit settings for project management properties.
  - **Portfolios:** View a list of portfolios.
  - **Template Config:** Open the project template configuration page.
  - **Workbench Config:** Open the workbench configurations page.

## Eureka and Prior Versions



The Project application (Eureka)

- **Getting started:** Access the wiki documentation for the Project application.
- **Project Overview:** Open the the Project Overview homepage, which contains several built-in reports.
- **Portfolios**
  - **Create New:** Create a new portfolio.
  - **All:** View all portfolio records.
- **Projects**
  - **Create New:** Create a new project.
  - **Top Level:** View all projects that do not have a parent project.
  - **Pending:** View pending projects.
  - **Work in Progress:** View projects currently in progress.
  - **Closed:** View projects already closed.
  - **All:** View all project records.
- **Project Tasks**
  - **Pending:** View pending project tasks.
  - **Work in Progress:** View project tasks currently in progress.
  - **Closed:** View project tasks already closed.
  - **All:** View all project task records.
- **Administration**
  - **Properties:** Edit settings for project management properties.
  - **Import Project:** Importing Projects from Microsoft Project.
  - **Export Project:** Exporting a ServiceNow project to be used in Microsoft Project.

# Integration with Project Portfolio Suite

Project Management can be used as a separate application or it can be activated as part of the Project Portfolio Suite (PPS). This application provides a simplified, team-oriented approach to IT development by combining several individual applications and integrating the different components of the project development lifecycle.

# Activating Project Management

Administrators can activate the Project Management plugin.

Upgrading to the Dublin release does not automatically upgrade you to the v3 application. If Project Management v2 is active and you want to upgrade to v3, read the upgrade instructions.

**Click the plus to expand instructions for activating a plugin.**

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

The Project application can also be activated as part of the Project Portfolio Suite.

# Enhancements

## Fuji

- The Project Management application is integrated with Project Portfolio Suite (PPS).
- Components of the SDLC (Scrum Process) are also integrated with Project Management to enable a project management approach that combines the Waterfall and Scrum methodologies.
- The project workbench provides a central location for managing projects and project phases. The workbench supports both the Project Management and Application Lifecycle Management applications, allowing for a hybrid approach to project management.
- The project calculation engine supports manual project calculation in addition to auto calculation.
- The composite field combines information from two different fields, typically a project or project task number and a short description.
- Project templates define the basic structure of a project and enable the project manager to create, save, and reuse project structure. This feature is available with the Fuji release.
- The IT Finance application adds a Finance view to the Project and Portfolio forms. The Finance view adds a chart that shows expenses that were allocated to the project or portfolio, shown by the financial bucket that the expense is associated with. See IT Finance for more information.

## Eureka

- Project managers can export ServiceNow projects to Microsoft Project, where the project can be managed and then imported back into ServiceNow.
- New business rules populate project-specific fields on non-project tasks with default project task data when these tasks are added to a project.
- The Project application automatically creates a new record in the Portfolio Project table for all new projects. This allows project managers to add a project to a portfolio by associating the newly created record with the portfolio.

## Dublin

- A new version of the Project application is available: version 3. See Project Management v2 to v3 Upgrade for upgrade information and instructions.
- Resource planning can be accomplished through the Resource Management application.
- The core project engine has been improved for the new version of the application. This results in better performance, usability, and scalability, especially with large projects.
- The Project application includes a default schedule that is applied to all new projects and project tasks. The schedule uses a 40-hour work week, from 8 A.M. to 5 P.M. with an hour break at noon.
- Project managers can now link existing change request records or create new change request records to link to project tasks. This feature links change management with project management.
- The Project application framework that supports importing from Microsoft Project 2010 was improved.
- State changes now roll down from the project to project tasks and from parent tasks to child tasks. For more information, see Project Task Relationships and Dependencies.
- Project managers can no longer modify most of the form fields on parent tasks. This enforces the concept that all parent tasks should derive aggregate values from their child tasks.
- A work breakdown structure (WBS) is available for project managers, and a new view, **WBS**, is available on the Project form.
- For v3, the functionality of the project management costing add-on has been moved to the Cost Management plugin.
- Projects can be included in multiple portfolios.

# Project Reporting

❗ **Note:** *This article applies to Fuji and earlier releases. For more current information, see Project Management*[1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.'**

## Overview

Get an at-a-glance view of projects with reports, the portfolio dashboard, and project views. See information such as projects with slipped milestones, graphs of resources by project, and projects listed by percentage complete.

❗ **Note:** *Much of information available on reports, dashboards, and views is customizable. The examples below derive from defauit settings.*

## Project Management Reports

ServiceNow provides several global reports, both lists and charts, that show the status of projects at a glance. You can also create custom reports or create reports that can be viewed by certain groups. The following project management reports are available in the base system:

- Active projects
- Active Projects by Manager
- Pending projects
- Projects (by priority)
- Projects (by risk)
- Projects (by state)
- Active Project Unassigned Tasks 30 Days

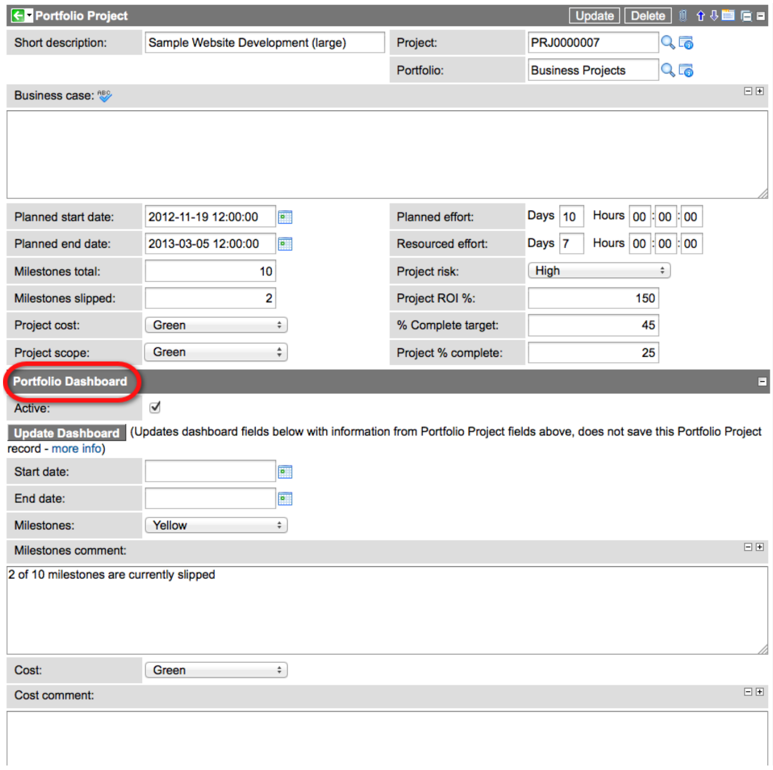
To access global reports, navigate to **Reports > View/Run**. See Introduction to Reports for more information.

## Portfolio Dashboard

Create a portfolio dashboard to summarize key information for all projects in a portfolio and add reporting data and details, such as the number of slipped milestones, that are not included in the project records or project task records. After creating the portfolio dashboard, add it as a gauge on a homepage, such as the Project Overview page.

### Creating a Portfolio Dashboard

1. Select **Project > Portfolios > All**.
2. Select a portfolio.
3. In the **Portfolio View** related list, click a project link. The Portfolio Project form opens, with the **Portfolio Dashboard** section at the bottom.



Portfolio Project

| | | | |
|---|---|---|---|
| Short description: | Sample Website Development (large) | Project: | PRJ0000007 |
| | | Portfolio: | Business Projects |

Business case:

| | | | |
|---|---|---|---|
| Planned start date: | 2012-11-19 12:00:00 | Planned effort: | Days 10 Hours 00 : 00 : 00 |
| Planned end date: | 2013-03-05 12:00:00 | Resourced effort: | Days 7 Hours 00 : 00 : 00 |
| Milestones total: | 10 | Project risk: | High |
| Milestones slipped: | 2 | Project ROI %: | 150 |
| Project cost: | Green | % Complete target: | 45 |
| Project scope: | Green | Project % complete: | 25 |

Portfolio Dashboard

Active: ☑

Update Dashboard (Updates dashboard fields below with information from Portfolio Project fields above, does not save this Portfolio Project record - more info)

| | |
|---|---|
| Start date: | |
| End date: | |
| Milestones: | Yellow |

Milestones comment:

2 of 10 milestones are currently slipped

Cost: Green

Cost comment:

**The Portfolio Project form, with the Portfolio Dashboard at the bottom**

The top part of the Portfolio Project form contains data inherited from the project record, such as project start and end dates, the ROI percentage, and the percent complete. The final, published dashboard uses this data together with the fields that you configure in the **Portfolio Dashboard** section of the form.

4. Fill in the following fields in the top part of the form:
   - **Milestones total:** total number of milestones for this project.
   - **Milestones slipped:** number of milestones missed to this point.
   - **Project cost:** subjective evaluation of the current cost of the project, represented by a color: **Green**, **Yellow**, or **Red**.
   - **Project scope:** subjective evaluation of the current scope of the project, represented by a color: **Green**, **Yellow**, or **Red**.
   - **% Complete target:** anticipated completion percentage for this project at this time. This is the target percentage against which the actual completion percentage is compared.
5. Click **Update Dashboard** to populate the fields in the dashboard with the data from the top part of the form.
6. Add any additional information, such as display colors and comments for the selections in the **Portfolio Dashboard** section of the form.
7. Click **Update**.

   This action updates any gauges in use for this portfolio.

# Updating a Project in Progress

## Overview

While a project is underway, keep actual values as current as possible. Continue to measure actual values, analyze the impact of any potential project risks, and make adjustments to handle scope, cost, and schedule accordingly. Also take advantage of two ServiceNow features that help project participants interact and collaborate: journal fields and live feed.

### Key Terms for This Topic

- **Baseline:** a collection of planned start and end dates for each task (as opposed to the actual start and end dates). Baselines provide points of comparison between planned dates and actual dates.
- **Journal fields:** input fields that can allow, store, or display input. Use journal fields on Project forms to keep notes on important project developments.
- **Live feed:** a ServiceNow tool that allows the posting and sharing of content. Enable live feed on project forms to allow project members to collaborate with each other by posting messages and uploading files.

## Update Project Fields

While the status of the project is **Work in Progress**, keep these fields up to date:

- **Priority:** the priority of the project, especially as it relates to other projects in the portfolio.
- **Net value:** the project's value to the company expressed in expected revenue.
- **Risk cost:** the sum of all costs involved with potential project risks. Although this field is not related to cost management, you can use this field to estimate the costs of risks that arise during the project.
- **Configuration item:** the CI related to the project.
- **Schedule:** the type of work schedule.
- **Work notes:** a useful record of notes and comments related to the project.
- **Live feed:** a record of the collaboration between various project stakeholders (requires Live Feed).

For project tasks, keep these fields up to date:

- **State:** remember to change project task states to **Work in Progress** when the task should begin (for tasks that have a specified start date) and **Closed** when the task is finished. Task states do not change automatically except when the task's time constraint is set to **Start ASAP** and the predecessor task's state is changed to one of the closed states.
- **Assignment group:** the group of resources currently working on the task.
- **Assigned to:** the individual assigned to the task.
- **Time cards:** the amount of time resources work on a project, which roll up into **Actual effort**. If a labor rate is configured for a time card, changes to the time cards affect the project's **Actual cost**.

# Track Project Comments and Collaborate with Stakeholders

ServiceNow provides the following features that enable you to save and track communication and interaction between project stakeholders.

## Journal Fields

Two useful journal fields that ServiceNow provides are **Comments** and **Work notes**. Configure any project form to show these fields.

## Live Feed

To enhance collaboration between various stakeholders, enable live feed. Live feed is essentially an archive of content that is related to a record. Live feed content can include written messages, just like comments, and file attachments, such as feature specs or design documents related to the project. When live feed is attached to a specific record, such as a project record, it is referred to as a document feed.

To add live feed to project forms:

1. Navigate to **System Definition > Dictionary**.
2. Click the **pm_project** table name that has no corresponding **Column name**.
3. In the **Attributes** field, enter **live_feed=true**.
4. Click **Update**.
5. Navigate to **System Definition > UI Actions**.
6. Open the **Follow on Live Feed** list action.
7. In the **Table** field, select the [pm_project] table.
8. Right-click the header and select **Insert** to create a copy of the UI action for the [pm_project] table.
9. Repeat steps 5-8 for the **Show Live Feed** form action.
10. Configure the Project form and add **Activities (filtered)** (the activity formatter) to the desired location on the Project form.

### Live Feed Project Notifications

The following Live Feed notifications are created for a project, starting with the Fuji release. To see these notifications, click the Live Feed icon ( ) on the project form header.

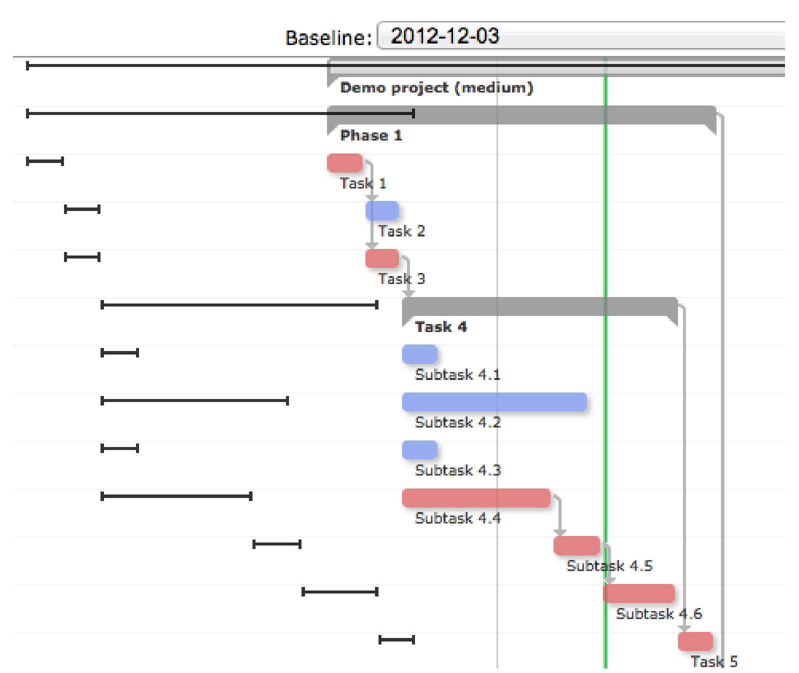
- A project task state is changed
- An issue or risk is created or an issue or risk state is changed
- A project baseline is created

For changes in the state of a project task, risk, or issue, the live feed message includes a link to the specific record. Click on this link to open the record in a popup window.

# View the Project Baseline

If the project is not on schedule, create a baseline to get an idea of how much schedule slippage has already occurred. A baseline is a collection of all planned dates for all tasks and milestones at the time you create the baseline. On the Gantt chart, the baseline appears as a set of black lines offset to the left or right of the bars that represent the actual tasks. The black lines represent planned dates, while the task bars represent the actual dates. Typically the project tasks are shifted to the right of the black baselines, which indicates that a project is running behind schedule.

Create a new baseline any time you want the most up-to-date view of deviations between actual and planned values. You can create as many baselines as necessary throughout the project. Baselines are available in the Calgary release.



**A baseline from a demo project**

**Note:** *Nested projects cannot have baselines. Baselines are allowed for top-level projects only.*

## Creating a Project Baseline

To create a project baseline:

1. Navigate to the Project form and click the **Create Baseline** related link. A dialog box appears.
2. Enter a name and a description.
3. Click **OK**.

## Opening a Project Baseline

To open a project baseline, do one of the following:

- Navigate to the Gantt chart for the project and select the baseline from the **Baseline** list at the top.
- Use the baseline related list:

  1. Verify that the **Baseline** related list is part of the Project form. If not, configure the form to add it.
  2. Open the baseline record from the related list to see the Baseline form, which has a **Baseline Items** related list. The related list contains the planned start and end dates for each task.
  3. On the Baseline form, click **View Baseline on Gantt chart**.

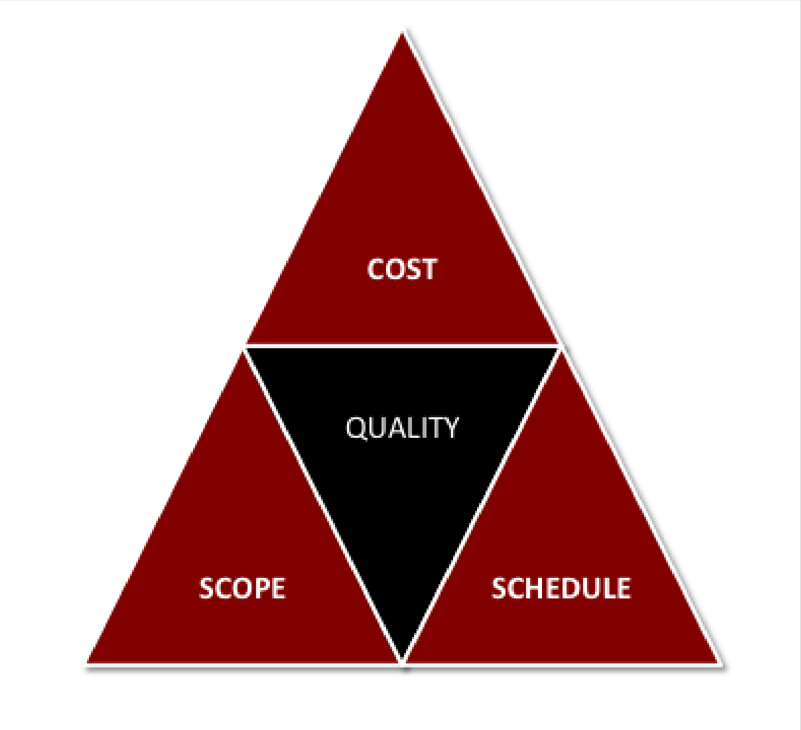
## Analyze Project Costs

See Managing Project Costs for more information on the project's actual cost.

## Options for Modifying Project Components

After you analyze the project timeline and updated costs, decide which aspects of the project triangle (scope, cost, and schedule) to modify. The following actions are things you might want to consider.

- Assigning more resources to existing tasks on the critical path to help meet a deadline.
- Creating additional project phases and tasks to cover an expanding scope.
- Increasing the number of work hours on the schedule to help meet a deadline.
- Eliminating overtime from the schedule to reduce costs.
- Reevaluating the dependencies of tasks to determine if any tasks that make up the critical path can run concurrently instead.



# Closing a Project

## Closing a Project

### Overview

Closing the project involves more than just changing the project state to closed. Post-project activities include viewing baselines and actual values to evaluate how the much the project slipped from its original estimates. They also include following up on related incidents, problems, or changes that are linked to the project through ServiceNow.

Normally, you should not reopen a project after it is closed. See Reopening or restarting a project for more information.

### Closing the Project

When you close a project, all project tasks are closed automatically in the v3 application, which is available starting with the Dublin release. See Project State Rollups and Roll Downs for details.

In the v2 application, projects do not close automatically when they reach the planned end date or when the final task on the critical path is closed. Likewise, the tasks in a project do not close automatically when they reach the planned end date or when the project is closed. You must change the state of all project tasks and the project to closed after determining that all work is complete.

To close a project:

1. Verify that the work is completed for all tasks in the project.
2. On the Project form, change the **State** field to one of the closed states:
   - **Closed Complete:** the project is finished and all tasks are complete.
   - **Closed Incomplete:** the project is finished, but tasks remain unfinished.
   - **Close Skipped:** the project was abandoned.

Alternatively, close every project task first, starting with the lowest-level child tasks. The closed states roll up to parent tasks, and when the highest-level parent tasks are all closed, the project state changes to closed. However, the default closed states for parent tasks and for the project is **Closed Complete**. This means that even if you change any or all project tasks to **Closed Incomplete** or **Closed Skipped**, the project state is changed to **Closed Complete**.

Also change the project's **Phase** field to **Closing**. The project phase is for reference only and is not linked to or dependent on the **State** field.

# Creating a Final Baseline and Viewing Actual Values

After the project is complete, create a final baseline to see how closely the actual project values came to the estimated values.

**Note:** *Baselines compare only planned start and end date values with actual start and end date values. Use reports to compare effort and cost.*

# Reusing a Project as a Template

If any future projects will be structured in a similar manner and use similar tasks or resources, copy the project and use the copy as a template. To copy the project:

1. Navigate to the Project form for the project to be reused.
2. Right-click the header bar and select **Copy Project**.
3. Enter the **New Project Name** and click **OK**.

Copying a project creates a new project record in the Project table with the name entered. The project number is automatically generated and the state fields of the project and all tasks are set to pending.

# Managing Related Incidents, Problems, and Changes

For projects created from an incident, problem, or change, updating the project state does not automatically update the related incident, problem, or change request record. You must update the related record manually. For example, if the completion of a project also means that a related change can be closed, go to the change record and modify its **State** field. It is also a good idea to update the work notes field on the related record to include any relevant information about the project.

# Project Costs

# Project Management v2 Costing Add-on

## Overview

The **Project Management v2 Costing Add-on Plugin** adds features that connect the Project Management Plugin to the Cost Management Plugin, to allow for estimating and tracking the costs associated with projects. After activating the plugin, see Managing Project Costs.

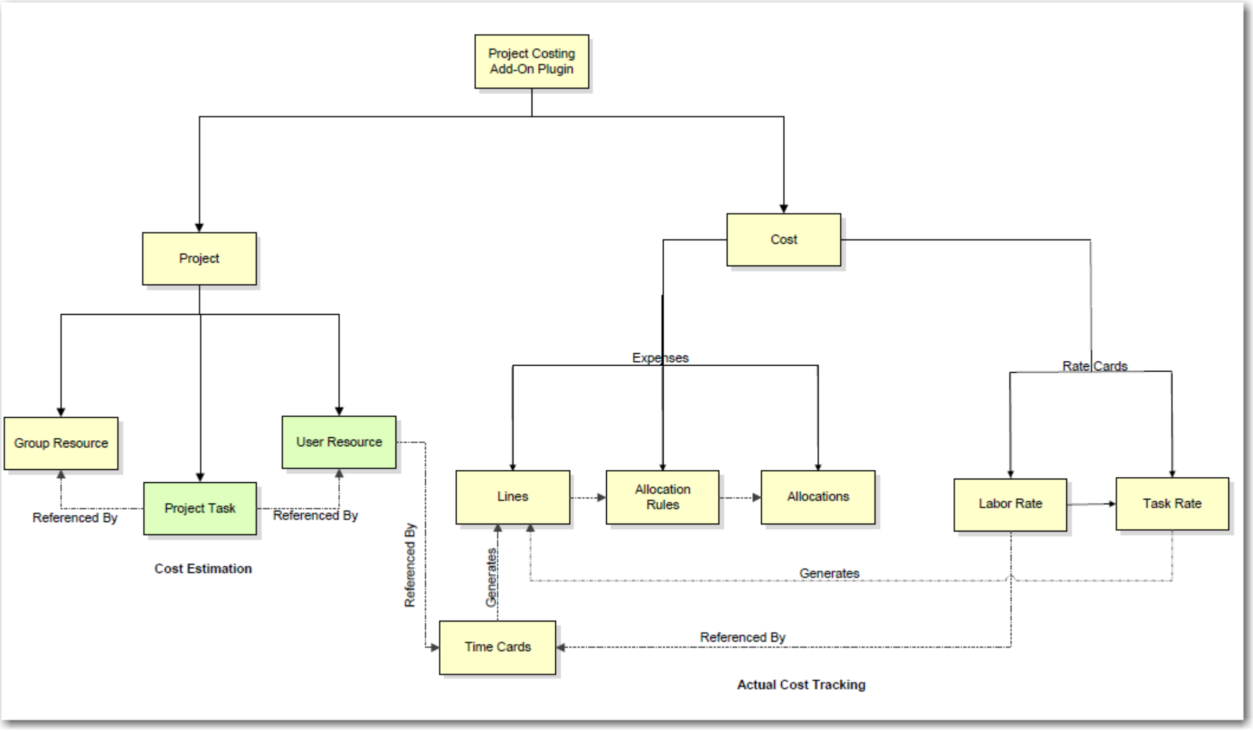
**Note:** *The functionality for the Costing add-on is built into the new Project v3 application starting with the Dubin Release.*

This plugin enables the following project costing features:

- Estimate **group resource** costs during project planning.
- Tracking the actual cost of each **user resource** for a project.
- Track actual project task costs from **time cards** and other project expenses.
- Allocate project costs to the business.
- Represent project costs to the project's affected CIs.
- Rollups of actual task expenses to parent tasks and the project record.

# Concepts

The concepts within the Project Managment v2 Add-on Plugin are the same as those within Project and Cost. The following diagram details how the concepts work together:



# Installed Components

## Properties

To access these properties, type **sys_properties.list** in the Navigation tree filter.

| Property | sys_property Name | Description |
|---|---|---|
| For planned tasks types, calculate the actual cost field using the total of expense lines for the task. | glide.cost_mgmt.calc_actual_cost | Value: **true/false**. Default: *true*. This property is from Cost Management. When an expense line is created against any task of **planned_task** type and this property is *true*, the system gets a sum of the costs for all the task’s expense lines and sets the total cost in the task’s **work_cost** field. |
| When creating a task expense line should the system also create expense lines for the task's top task? | glide.cost_mgmt.process_task_top_task | Value: **true/false**. Default: *true*. |
| Enable project cost rollup (estimated and actual) - updating the cost of a project task will update the cost of its parent | com.snc.project.rollup.cost | Value: **true/false**. An existing property is set to true with this plugin. |

## Business Rules

The following business rule is used by this plugin:

- **Project Cost Rollup [planned_task]:** This business rule is installed with the Project Management plugin. It calculates and rolls up project cost.

The following business rules are modified by this plugin:

- **Calculate Group Estimated Cost [group_resource]:** Calculates estimated group resource cost using the group's hourly rate and number of hours estimated.
- **Process Top Task Parent [fm_expense_line]:** For task source expenses, determines if a duplicate expense for the task's top task should be created.
- **Update User Resource Cost [user_resource]:** Calculates the users actual cost for a given number of actual hours. The cost is calculated using labor rate cards for the user. If no labor rate cards can be found for the user, the default rate is used from property `com.snc.time_card.default_rate`. See Managing costs for more information.

# Getting Started

There is no need to activate this plugin for new instances starting with the Dublin release. For instances upgrading from an earlier release, administrators need to activate the plugin.

## Activating the Plugin

**Click the plus to expand instructions for activating a plugin.**

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

# Managing Project Costs

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Project Management Costing Add-On* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

Once the Project Management v2 Costing Add-on Plugin is installed, the costs associated with a project can be estimated before-hand, and actual costs can be tracked as they are recorded. Actual costs can then be associated to the Business Services affected by, or driving the project, and then costs can be allocated to Cost Centers.

## Estimating Cost

There are two methods of estimating project costs, based on what costs are being estimated:

- Labor Costs - estimated using **Group Resources**.
- Task Costs - estimated using the **Estimated Cost** field on the Project and Project Task form.

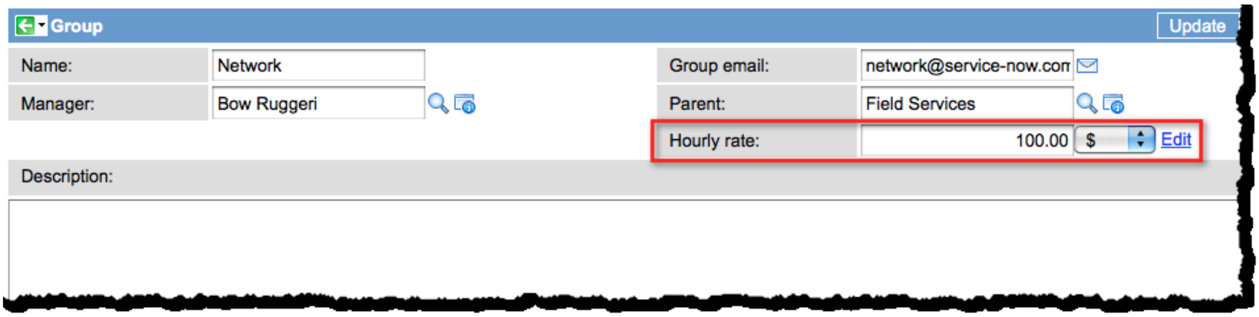
### Estimating Labor Costs

Labor costs are estimated using the **Group Resources** related list.

The plugin creates an **Hourly rate** field on the Group form and adds a new form called **Group resource**, which calculates the estimated cost for the group, based on the hourly rate provided. The **Group Resources** will be added automatically based on the groups associated with the Project or its associated Project Tasks.

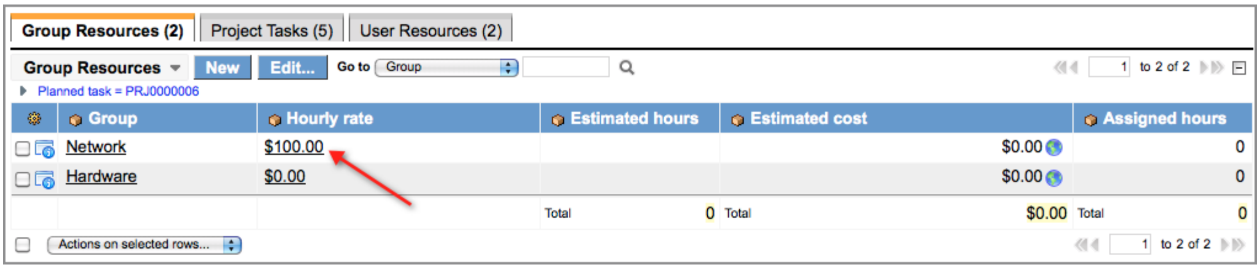
**'To estimate the Project's cost using Group Resources:**

1. Select a group and configure an **Hourly rate** using the new field in the Group record.



| | | | |
|---|---|---|---|
| Name: | Network | Group email: | network@service-now.com |
| Manager: | Bow Ruggeri | Parent: | Field Services |
| | | Hourly rate: | 100.00 $ |
| Description: | | | |

2. Click **Update**.
3. In the Group Resources Related List, click the link on the newly configured **Hourly rate**.

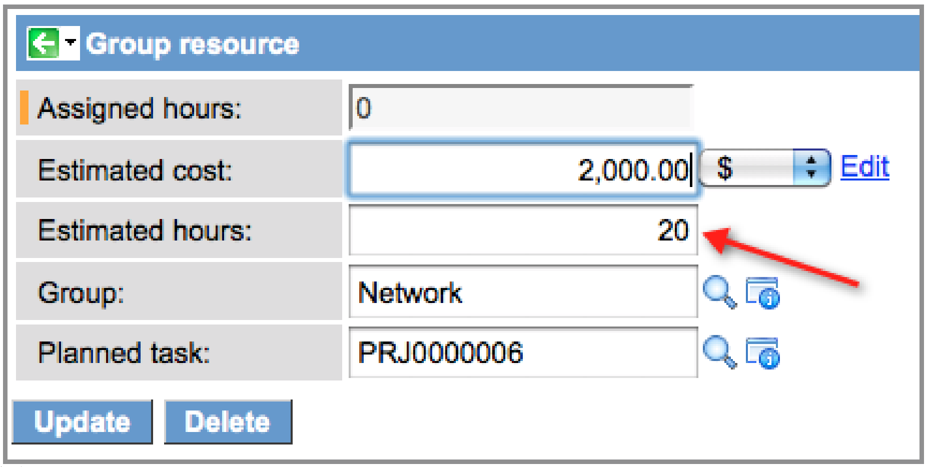


Planned task = PRJ0000006

| Group | Hourly rate | Estimated hours | Estimated cost | Assigned hours |
|---|---|---|---|---|
| Network | $100.00 | | $0.00 | 0 |
| Hardware | $0.00 | | $0.00 | 0 |
| | | Total 0 | Total $0.00 | Total 0 |

The **Group resource** form appears.

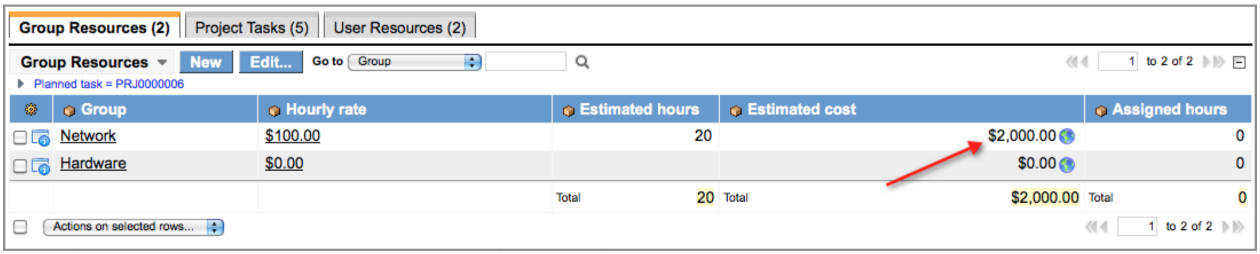
4. Configure the **Estimated hours** this group will need to complete the project. **Note:** if the group has already been assigned to the lowest level sub-tasks, and those sub-tasks have an **Planned Effort** field (visible in the advanced view) populated, then the **Assigned Hours** field will total all of the time currently assigned to this group resource. Use this as a guideline as the **Estimated Hours**.



| | |
|---|---|
| Assigned hours: | 0 |
| Estimated cost: | 2,000.00 |
| Estimated hours: | 20 |
| Group: | Network |
| Planned task: | PRJ0000006 |

5. Click **Update**.

   The Costing Add-on plugin calculates the **Estimated cost** for this group and displays the results in the **Group Resources** Related List.



Planned task = PRJ0000006

| Group | Hourly rate | Estimated hours | Estimated cost | Assigned hours |
|---|---|---|---|---|
| Network | $100.00 | 20 | $2,000.00 | 0 |
| Hardware | $0.00 | | $0.00 | 0 |
| | | Total 20 | Total $2,000.00 | Total 0 |

6. Repeat this process for the other group resources in the project.

## Estimating Task Costs

Task costs can be estimated using the **Estimated Cost** field which can be added to the Project and Project Task forms.

**To estimate task costs:**

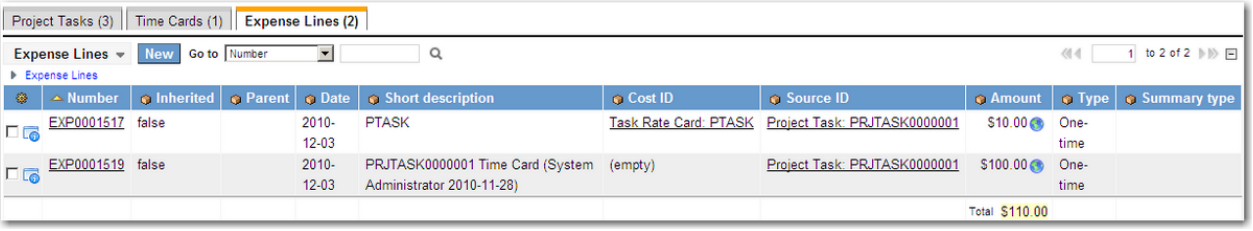
1. Enable the Estimated Cost field calculations.
   1. Navigate to **Project Management > Properties**
   2. Set the property *Enable project cost rollup (estimated and actual) - updating the cost of a project task will update the cost of its parent* to **Yes** (true).
2. Navigate to the lowest level sub-tasks of the project and populate the *'Estimated Cost* fields.

Once those fields are set, the top-level project's **Estimated Cost** field should reflect the sum of all the lowest level sub-tasks' estimated costs.

# Tracking Actual Costs

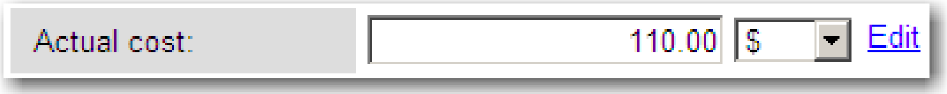
Once the project is planned, actual expenditures can be tracked while the project is underway. To track costs, define rate cards for the task and labor expenses. These rate cards will automatically generate expense lines showing actual expenditures, which will be associated with the projects.

If rate cards are defined, the Task Expense Lines will be generated as each Project Task closes, and Labor Expense Lines will be generated when Time Cards are approved. These are visible in the **Expense Lines** related list (advanced view). This screenshot has one expense line incurred by a **Task Rate Card** and one generated by a Time Card associated with the project:



| | Number | Inherited | Parent | Date | Short description | Cost ID | Source ID | Amount | Type | Summary type |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | EXP0001517 | false | | 2010-12-03 | PTASK | Task Rate Card: PTASK | Project Task: PRJTASK0000001 | $10.00 | One-time | |
| □ | EXP0001519 | false | | 2010-12-03 | PRJTASK0000001 Time Card (System Administrator 2010-11-28) | (empty) | Project Task: PRJTASK0000001 | $100.00 | One-time | |
| | | | | | | | | Total $110.00 | | |

These expenses are totaled in the tasks' **Actual Cost** field:



Actual cost: 110.00 $

And the parent tasks will add these costs into their own **Actual Cost** fields.

## Allocating Costs

Project costs can be allocated to a **Cost Center** using Expense Allocation Rules. For more information, see Expense Allocation Rule.

# Cost Rollups

Costs are rolled up from child tasks to parent tasks all the way up to the project. See Parent-Child (Rollup) Task Calculations for more information.

## References

[1] https://docs.servicenow.com/bundle/jakarta-it-business-management/page/product/project-management/concept/c_ProjectManagementCostingAddOn.html

# Reference

# Project Management FAQ

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Project Management* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Fuji Release

### How can I create new portfolio project views for a portfolio which we were able to do earlier in Eureka?

You can do so by clicking on the **Portfolio Status** related link on the Portfolio form and then clicking **New**.

### Why don’t I see Agile or Test Phase selection on project workbench?

You may not have activated the Project Portfolio Suite plugin. If only Project Management is activated, then you can create only the Waterfall phase.

### Why can’t I create more than one Agile phase, Team or Test phase?

This is a known limitation in the Fuji release. This will be addressed in a future release.

### Why can’t I change my auto project to manual?

If an auto project has tasks, it cannot be converted to a manual project.

### Why can’t I have both auto and manual tasks in a manual project?

This is a known limitation in the Fuji release. This will be addressed in a future release.

## Why does it take so long to activate Project Portfolio Suite?

It can take a long time to activate this application if you have a lot of task records or if you are installing the demo data.

## I saved a template but it doesn't show up in the Project Templates List

To save a project as a template, click the **Save as New Template** related link on the Project form. Right-clicking the form header and selecting **Templates > Save as Template** does not create a project template.

## Why can't I overwrite certain project or task fields (for example, Rollup or Schedule) under the project template?

These certain fields are not allowed to be intentionally overwritten.

## Does the copy project feature copy the stories in an Agile phase and the test cases in a Test phase?

Currently this feature only copies the project tasks.

## Why don’t I see some of the Project menu options in the application navigator? For example, Workbench or Agile?

Some of the menu options are only available to users with the project manager role.

## Why don’t I see the Accept button on the Idea form?

The **Accept** button is available to users with the demand manager role.

## Why don’t I see the other demand types, such as Defect or Enhancement?

These additional demand types are available if the Project Portfolio Suite or Software Development Lifecycle (SDLC) plugins are activated.

## In the visual task board, why can't I move a card from one lane to another?

This can happen if you have not filled in all of the mandatory fields.

### Why can't I sort on the first field in a composite field?

Sorting is currently supported based on the second field in a composite field.

### Why can't I sort on the Stage column on the Demand form?

Sorting on the Stage field is not currently supported.

## Eureka and Prior Releases

### Can you select a calendar in Project Management v2 that enables you to state non-working time?

Yes. Each project has a **Schedule** field on it, allowing you to select a working calendar (such as **8-5 M-F**).

### Does this module use PMI processes (best practices)?

Project Management v2 was developed based on a combination of standards (Prince2, PMI, Agile), as well as on an abundance of customer input and feedback.

### Can you assign resources to multiple tasks at once and not just one at a time?

No. The current implementation permits one assignment at a time. To expedite this process, you can disable the confirmation pop-up.

### Do you differentiate between "Requests" that are having analysis done on them versus Projects?

Yes. The **State** field on a project can be used to distinguish projects that are simply being considered from projects that have moved into the Project Management process. The default **Pending** state might be sufficient, but if you want to add a selection, right-click on the **State** field and select **Personalize choices**. Add the choice **Under consideration** to the list.

### Can you print out the Gantt Chart?

Yes, but the output is limited to the browser's printing capability.

### Can you show dependencies between projects?

Yes. Multiple projects can be part of a larger project, and their dependencies will be displayed just like those of the project task.

### Do resources for project management need a license?

Project Management follows the same licensing model as the rest of the product. Process users (admins and task assignees) need a license, and everyone else is free.

## Can we create a Product Management application to keep track of feature requests?

Yes. Due to the schema design, you can extend the *planned_task* table with a new table called *feature_request*. The new table will function exactly like *project_task* table.

## Is there a way that a project task can begin prior to completion of a predecessor project task?

Yes . . . and No. Every task can optionally be set to a **Start on a specific date** time constraint, allowing you to fully control when it starts. However, a task *cannot* be set to start prior to its predecessor's end date. If you choose to do so, the predecessor relationship will be broken automatically.

## Does the Project Management v2 Plugin do resource leveling?

Not yet. Resource data is now being collected, and time lines have been provided to make manual resource management easier. Automatic resource leveling is something that may be available in a future release.

## Can you prioritize tasks, either within a project or between types of work like project tasks, incidents, and so on?

Yes. All project tasks, incidents, problems, change requests, catalog requests, and release tasks go into the same work queue (*Service Desk* > *My Work*), and can be sorted by priority (or any other field you want).

## Does this application replace Microsoft Project?

Yes. A utility is included with Project Management v2 that enables you to import existing Microsoft Project projects directly into ServiceNow.

## Can you add more fields to the Propose a New Project form?

Yes. The Propose a New Project form is nothing more than a fully customizable record producer.

## What is the minimum granularity for tasks in the Gantt Chart? Can you have a 5 minute task?

There is no minimum time limit. A 5 minute task is allowed, but depending on your zoom level, may display slightly larger than it should. We have a minimum span pixel width of 10px (4px for the left drag, 4px for the right drag, and 2px for the X/Y moving).

## Are line managers able to approve when resources are assigned to a project by a project manager?

Yes, because of the ability to associate a workflow with any object in the system, and have that workflow initiate any number of approvals. There are many options for creating additional processing.

## Can a submitted time card go through an approval by the project manager?

Yes. See above.

## Are other tasks (like Change) shown in the resource usage view?

Not be default, but they can be. The Gantt Chart is designed to be easily modified to display anything you want. Anything with at least one date can be placed on a Gantt Chart.

## Can we extend the resource chart to tasks other than Project?

Yes. See above.

## Can Project be exported from ServiceNow to Microsoft Project?

No. We've built the import tool, but the export tool will have to wait till a later release.

## Can Project be related to Change?

Yes. Since Project and Change are both derivatives of the *task* table, they can relate to one another as easily as Incident and Problem do.

## Can the Project and portfolio management suite also be used for an employee time tracking system? Can employees log the number of hours they worked on a project in the system?

Yes. The time cards can be used to track any hours against any task in the system, including Projects.

## Can we associate schedules with resources?

Not yet. A resource's schedule would have to include all of their calendar data (meetings, time off, working hours, etc) to be meaningful. Until we have a system in place to integrate with MSExchange and Lotus Notes easily, the schedule will remain at the Project level.

## Can we get alerts when the resource is allocated for more than 100%?

For a single project, a user cannot be allocated more than 100%. The system simply doesn't allow it. As for their allocation across all projects, theoretically you could calculate their resource allocation, but that involves timing. For instance, if a user is allocated 80% to a Jan-Jun project, and 80% to a May-August project, is that user really over 100% allocated? Depending on where his 80% time is spent in either project, he may or may not be over 100%. At this point, the system is just collecting the data on a per-project basis.

### Can the resource chart display the unavailability of a resource, such as when he is on vacation or sick?

Yes. There is no specific place to put time off data, but you could create a PTO project and allocate some of a user's time to it.

### Can we import and export Microsoft Project Plan to ServiceNow Project and the Service Portfolio Management tool?

The **Import Project** module supports importing **.mpp** files, automatically converting the project to a ServiceNow project. Import Sets enable importing information from other file-types (**.xml**, **.xls**, and **.csv**) with the use of custom transport maps.

You can also export projects in XML format that you can use in Microsoft Project. See Importing and Exporting Projects for more information.

# Project Management Processing Rules

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Project Task Relationship and Dependencies* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

Project management enables you to create child tasks that are nested under a parent task and successor tasks that are dependent on the completion of a predecessor task. This page explains how to create such relationships and dependencies.

To create and edit portfolios, projects, and tasks, users must have the project_manager role in their user profile record. See Installed with Project Management for more information on roles in the Project application.

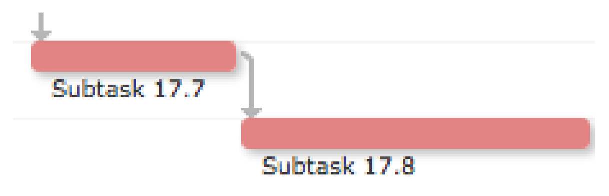
### Key Concepts and Terms for This Topic

- **Predecessor task:** a project task that, upon completion, is followed by another task. A predecessor task has a dependent relationship with its successor task.
- **Successor task:** a project task that cannot start until another task finishes. The successor task has a dependent relationship with its predecessor task.
- **Lag time:** a manually specified time break between predecessor and successor tasks. Configure a lag time, if necessary, when creating a predecessor-successor dependency.

  The system applies the project schedule in the v3 application (available starting with the Dublin release). For the v2 application, the system converts lag time to hours and does not consider the project schedule when applying lag time.
- **Parent task:** a project task with smaller tasks, referred to as child tasks, underneath it. Child tasks break down the work of a parent task into more manageable subsets. Certain fields for child tasks, such as planned end date, roll up and affect the same field in the parent task.
- **Child task:** a project task that is a subset of a larger task. Child task start dates cannot occur before the start date of the parent task.

- **Rollup task:** another term for a parent task in the context of aggregating child task items, such as effort or resources, into a larger parent task calculation. All fields on rollup task forms are read-only in the v3 application.
- **Roll down:** state changes roll down from the project to project tasks, and from parent tasks to child tasks in the v3 application.

# Task Dependencies



A finish-to-start dependency

A task dependency is created when one task is forced to start after another task finishes. For example, subtask 17.8 can only start when the **State** field of subtask 17.7 changes to **Closed**. The Project application only supports this kind of dependency, referred to as a finish-to-start dependency.

## Creating a Task Dependency

The easiest way to create a task dependency is with the Gantt chart.

Another option is to use related lists to create dependencies:

1. If the successor task does not already exist, navigate to the project form and create it.

   Do not create the task from the predecessor task form. Doing so creates a parent-child relationship.
2. Navigate to the predecessor task.
3. Configure the related lists for the Project Task form and add **Planned Task Relationship > Parent**. Do not select **Predecessor of** or **Successor of**.

   This adds the **Planned Task Relationships** related list to the Project Task form. This related list shows successor tasks.
4. In the **Planned Task Relationships** related list, click **New**.
5. On the Planned Task Relationship form, click the lookup icon and select the appropriate successor task.
6. Verify that the relationship **Type** is **Predecessor of::Successor**. *Do not* change this relationship type.
7. Enter the **Lag** time, if any, in either days or hours.
   - In the v3 application, the lag time takes the project schedule into consideration. If the lag time is 10 hours and the default schedule of an 8-hour work day is in use, the lag time pushes the task to the following day to cover the additional hours.
   - In the v2 application, the system converts the lag time to total hours. Therefore, a lag value of **1** day is equivalent to 24 hours. The lag time does not take the project schedule into consideration.
8. Click **Submit**.

## Task Time Constraints

The Project Task form includes a the **Time Constraint** field: either **Start ASAP** or **Start on specific date**.

- If the successor task is set to **Start ASAP**:

  The successor task appears on the Gantt chart as starting immediately after the predecessor completes without any lag time. However, the successor task can start on a later date if it has a value in the **Lag** field. To enter a lag value, double-click the relationship line in the Gantt chart and enter a **Lag** value. Alternatively, open the the predecessor task and enter a **Lag** value for the successor task in the **Planned Task Relationships** related list.

- If the successor task is set to a **Start on Specific Date** that is *later* than the finish date of the predecessor:

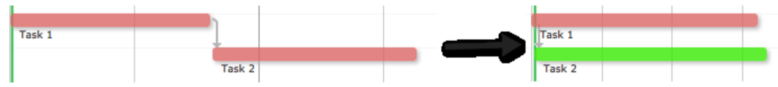
  The successor task starts at the time specified. On the Gantt chart, a lag appears just as if you set the **Lag** value on the relationship. However, the actual **Lag** value is not actually modified.

- If the successor task is set to a **Start on Specific Date** that is *earlier* than the finish date of the predecessor:

  ServiceNow changes the successor task time constraint to **Start ASAP** and the task starts immediately after the predecessor finishes, unless a **Lag** value exists.

## State Changes on Tasks in Dependencies

Dependencies do not affect the ability to change the state of predecessor or successor tasks. For example, if a project is already in progress, you can still change a successor task to **Work in Progress** even if the predecessor task has not finished. Also modify the successor task to start on specified date that is earlier than the planned end date of the predecessor. Although this would violate the dependency for planning purposes, ServiceNow provides this kind of flexibility in modifying the project. You can also perform actions like closing a successor task, and then opening a predecessor task. Although you are allowed to make these kinds of modifications to predecessors and successors, the related project tasks and the way they are represented in the Gantt chart might show unexpected results.



A successor task prematurely changed to Work in Progress

## Modifying Dependencies

To modify an existing dependency from the Gantt chart:

1. Double click the relationship line in the Gantt chart.
2. Enter a different task in the **Successor** task field.

To modify an existing dependency from a related list:

1. Open the Project Task form for the successor task and click the existing predecessor task in the **Planned task relationships** related list.
2. Enter a different task in the **Successor** field.

## Removing Dependencies

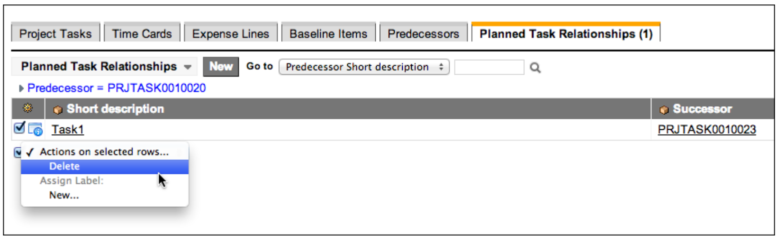
Remove a dependency that is no longer necessary from either the Gantt chart or the Project Task form. Removing the dependency also deletes the dependency record in the Planned Task Relationship table.

To remove a dependency from the Gantt chart:

1. Double-click the relationship.
2. In the **Planned Task Relationship** form, click **Delete**.

To remove a dependency from the Project Task form:

1. Open the predecessor task in the Project Task form and go to the **Planned Task Relationships** related list.
2. Select the check box beside the relationship being removed.
3. On the **Actions on selected rows** menu, select **Delete**.


Project Tasks
Time Cards
Expense Lines
Baseline Items
Predecessors
Planned Task Relationships (1)
Planned Task Relationships
New
Go to
Predecessor Short description
Actions on selected rows...
Delete
Assign Label:
New...

Predecessor = PRJTASK0010020

| Short description | Successor |
|---|---|
| Task1 | PRJTASK0010023 |

**Deleting the relationship record**

# Parent-Child Task Relationships

If a task is relatively large and requires several users with different skills to manage, break the task into subtasks and create parent-child relationships. A child task should be a relatively smaller, manageable size of work. When you group child tasks together under a parent, values such as **Estimated cost** aggregate and roll up to the parent task. So the parent task takes on the form of a *summary task* or *rollup task* for its child tasks. **Planned start date** and **Planned end date** rollup occurs when you create child tasks: the duration of the parent automatically adjusts to *cover* its child tasks.

A parent-child relationship is different from a dependency relationship. In a dependency, one task must finish before another begins. In a parent-child relationship, any number of tasks can be nested under a parent task with or without any dependencies. When you create a parent-child relationship, the parent task number is saved in the **Parent** field in the Project Tasks table. All project management tasks have a parent: either another project task or the project itself.

## Creating a Parent-Child Task Relationship

The easiest method is to create parent-child relationships is on the Gantt chart.

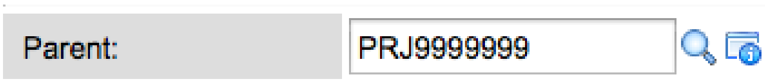
To create parent-child relationships with related lists:

1. Navigate to the parent task in the relationship.
2. In the **Project Tasks** related list, click **New**.

   The same Project Task form appears for all tasks regardless of the parent-child relationship.
3. Create the task and click **Submit**.

   The newly created task becomes the child task in the relationship.

To help remember what the parent of any task is, view the breadcrumb at the top of the Project Task form. It is also helpful to configure the form layout to include the **Parent** field.



Parent: PRJ9999999

**The Parent field. In this example the project is the parent.**

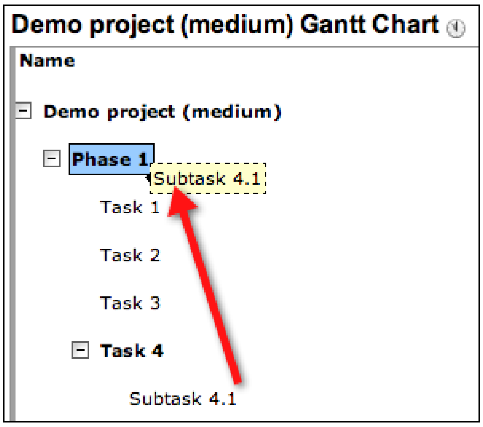
## Changing the Parent Task

To change the parent of a child task:

1. Navigate to the child task in the relationship.
2. Configure the form to add the **Parent** field, if needed.
3. In the **Parent** field, select the new parent task for this child task. To have the task stand alone in the project, select a project instead of a task.

## Modifying Parent-Child Relationships

To modify a parent-child relationship by using the Gantt chart, drag a child task to any other task (no matter whether it is a child or parent task). If you drag a child task up to the project name, it becomes a standalone task and is no longer considered a child task.



**Moving a child task**

To modify a parent-child relationship from a related list:

1. Navigate to the child task in the relationship.
2. Configure the form to add the **Parent** field if needed.
3. In the **Parent** field, select the new task that you want the child task assigned to. To have the task stand alone in the project, select a project instead of a task.

Unlike a dependency, a parent-child relationship is not saved as a record in any table. The only modification that takes place when a parent-child relationship is modified is the **Parent** field in the child task record.

## Time Constraints in Parent-Child Relationships

- If a child task is set to **Start ASAP**, the child task starts at the same time the parent task starts (as long as it does not have dependencies with other child tasks).
- If the parent task is set to **Start ASAP** and child tasks are set to **Start on specific date**, the earliest child task start date determines the start date of the parent (assuming no other dependencies). In this case, the parent's **Time constraint** field remains **Start ASAP** but the actual start date is changed to the start date of the earliest child task.
- If both the parent and first child task are set to **Start on specific date** but the first child starts later than the parent, the parent start date remains **Start on Specific Date** but the actual start date is pushed to the start date of the child. For example, if the parent task starts on October 1 and the earliest child task starts on October 2, the **Planned start date** of the parent is changed to October 2.
- Child precedence also applies to end dates. If the child task's estimated end date is later than the parent task's end date, the parent task's estimated end date extends to cover the child. For actual values, a parent has the same start date as the earliest start date of its children and the latest actual end date as the latest end date of its children, assuming the child tasks are **Closed Complete**. If the child tasks are not in the **Closed Complete** state, the actual end date of the parent is empty.
- For the planned start date of the parent task:
  - The planned start date is the earliest planned start date of all the children that do not have an actual start date.
  - If all child tasks have actual start dates, the parent task's planned start date is set to the actual start date.
- For the planned end date of the parent task, the latest planned end date or actual end date of the child tasks determines the parent's planned end date.

### Dependencies with Parent or Child Tasks

Options exist to create predecessor-successor relationships between child tasks with different parents, between two different parent tasks, or between a child task and another parent task. However, if the predecessor task finishes after the successor task starts, creating a dependency between child tasks that have different parents is not allowed.

## Parent-Child (Rollup) Task Calculations

Rollups involve date changes, state changes, and value calculations.

- Date changes involve modifying the planned start or end date of a parent task based on those values in child tasks.
- State changes involve modifying the state of the project record or parent task records if all child records are set to a certain state.
- Calculations involve summing the values of child tasks and then automatically updating the parent to reflect a new total.

Rollups work differently on these fields in the v3 application (available starting with the Dublin release):

- **Planned Start date:** set to read only for parent tasks. Remains editable for the project record (also considered the top-level task).
- **Planned End Date:** becomes read only.
- **Planned Duration:** becomes read only.
- **Actual Start Date:** becomes read only.
- **Actual end date:** becomes read only.
- **State:** becomes read only.

### Duration Rollups

Rollups are calculated for the following:

- **Planned duration and planned effort:** the sum of all planned duration and planned effort values for all child tasks.
- **Actual duration and actual effort:** the sum of all actual duration and actual effort values. Actual duration and actual effort values are calculated when all child tasks are in the **Closed Complete** state. Actual effort values can include rollups from time cards.

**Note:** *Verify that the time card property **Update the task's 'Actual effort' based on the hours entered in the time card** is enabled. Navigate to **Time cards > Administration > Properties** to enable this property.*

### Cost Rollups

Cost calculations roll up when the the Project Management Costing Add-on is active.

- **Estimated cost:** the sum of all cost estimates at the beginning of a project. Estimated costs of child tasks roll up to parent tasks and to the project.
- **Actual cost:** by default for the project, the sum of all costs of all the project's expense lines, which are typically associated with a a time card and a labor rate. To track costs, define rate cards for the task and labor expenses. These rate cards automatically generate expense lines showing actual expenditures, which are associated with the projects. If rate cards are defined, the task expense lines are generated as each project task closes, and labor expense lines are generated when time cards are approved. Expense lines are visible in the **Expense Lines** related list, which requires the **Advanced view** on a both Project and Project Task forms.

For actual costs of child tasks to properly roll up to the project and be added to project expense lines, the following must be true:

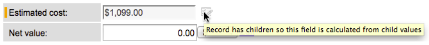
- The com.snc.project.rollup.cost property must be set to **true**. To enable this property, navigate to **Project > Administration > Properties** and select the **Enable project cost rollup** check box.
- The glide.cost_mgmt.process_task_top_task property must be set to false. To enable this property, navigate to **Financial Management > Admin > Properties** and select the **When creating a task expense line should the system also create expense lines for the task's top task** check box.
- The glide.cost_mgmt.calc_actual_cost property must be set to true. To enable this property, navigate to **Financial Management > Admin > Properties** and select the **For planned tasks types, calculate the actual cost field using the total of expense lines for the task** check box.

## Enabling Cost Rollup Calculations

To use rollup calculations:

1. Navigate to **Project > Administration > Properties**.
2. **Select Enable project cost rollup** and click **Save**.

Rollup values are read-only on forms. Point to the icon beside the field for a tooltip message.



Estimated cost: $1,099.00

Net value: 0.00

Record has children so this field is calculated from child values

**A rollup task for an estimated cost. The field is not directly editable on the parent.**

## Project State Rollups and Roll Downs

Project task states roll up. The state of parent tasks becomes read only in the v3 application, and changes automatically when you change the states of child tasks.

Project task states can roll up if:

- The state of the child task is manually changed and there are no other conditions on the parent task.
- The state of the child task is changed to **Work in Progress** or **Closed**. These states roll up to the parent. **Pending** and **Open** do not roll up to the parent task.

Project states can also roll down in the v3 application. If you change the state of a project to closed, all tasks under it change to the default closed value (**Closed Complete**). If a closed project or closed task is reopened, all tasks under it change as follows:

- Project or parent changed from closed to **Pending** or **Open**: child tasks change to **Open**.
- Project or parent changed from closed to **Work in Progress**:
  - Child tasks with a **Start on** date that has passed are changed to start **ASAP** and the state is changed to **Work in Progress**.
  - Child tasks with a **Start on** date that has not yet passed retain the same start on date but the state is changed to **Open**.

# Related Topics

# Schedule Pages

**Note:** *This article applies to Fuji and earlier releases. For more current information, see Schedule Pages* [1] *at* http://docs.servicenow.com **The ServiceNow Wiki is no longer being updated. Visit** http://docs.servicenow.com **for the latest product documentation.**

## Overview

A schedule page is a record that contains a collection of scripts that allow for custom generation of a calendar or timeline display. For a discussion of how calendars are created using Schedule Pages, see **Creating Calendars with Schedule Pages**. For information about custom generated timelines using schedule pages, see **Timelines**. See **Application Programming Interface (API)** for links to classes and methods in the API.

## Schedule Pages Form

To access Schedule Pages, navigate to **System Scheduler > Schedules > Schedule Pages**. The form provides the following fields, depending upon the **View Type** selected:

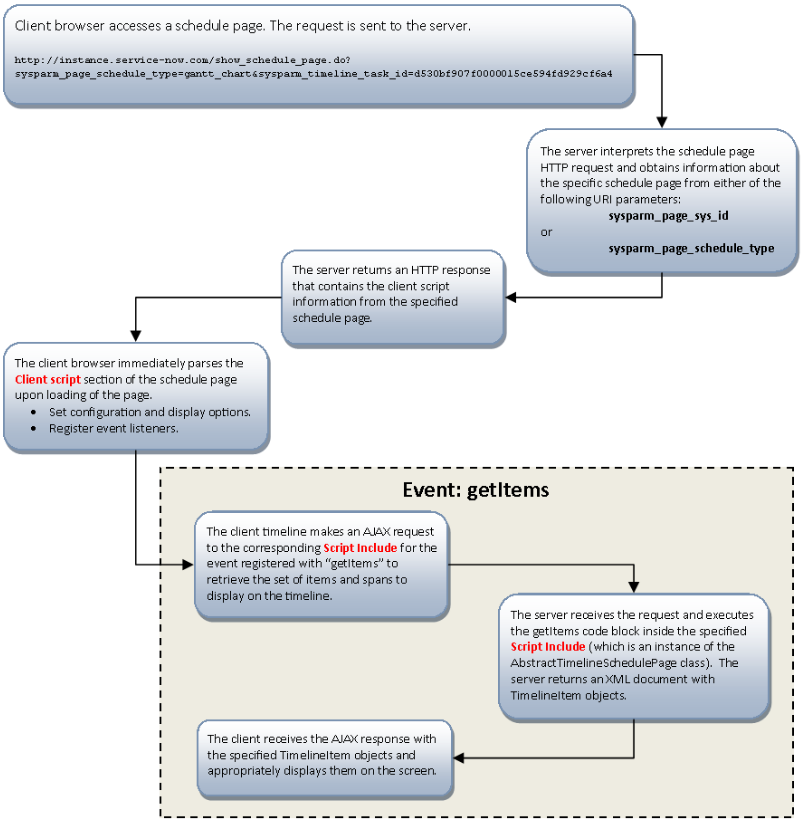
| Field | Field Type | Description |
|---|---|---|
| Name | String | General name that is used to identity the current schedule page. |
| Schedule type | String | The schedule type is a string that is used to uniquely identity the schedule page via the "sysparm_page_schedule_type" URI parameter. For example, a schedule page could be accessed as follows:<br>/show_schedule_page.do?**sysparm_page_schedule_type**=gantt_chart&sysparm_timeline_task_id=d530bf907f0000015ce594fd929cf6a4<br>Alternatively, the schedule page can also be accessed by setting the "sysparm_page_sys_id" URI parameter to that of the unique 32 character hexadecimal system identifier of the schedule page. |
| View Type | Choice | Each view type displays different field combinations. There are two options available :<br>- **Calendars**<br>- **Timelines** |
| Description | String | General description that provides additional information about the current schedule page. This field is not necessary. |
| Init function name | String | **Note:** *This functionality is only used by* ***Calendar*** *type schedule pages.*<br>The init function name specifies the name of the JavaScript function to call inside the **Client script** function for calendar type schedule pages. |

| | | |
|---|---|---|
| HTML | String | **Note:** *This functionality is only used by* ***Calendar*** *type schedule pages.*<br>The HTML field is a scriptable section that is parsed by Jelly and injected into the display page prior to the rest of the calendar. It can be used to pass in variables from the server and define extra fields are necessary. |
| Client script | String | The client script is a scriptable section that allows for configuring options of the schedule page display. The API is different depending on the schedule page view type and is discussed below. |
| Server AJAX processor | String | **Note:** *This functionality is only used by* ***Calendar*** *type schedule pages.*<br>The Server AJAX processor is specific to calendar type schedule pages that is used to return a set of schedule items and spans to be displayed. |

# Timelines

A Timeline Schedule Page is a specific record that contains configuration information for displaying time based points and spans in a "timeline" like fashion. The timeline schedule page references a script include that extends from `AbstractTimelineSchedulePage` to perform dynamic modification to the timeline based on different events and conditions. Both the schedule page and the script include for timeline generation allow for extreme customization and their corresponding application programming interface (API) is documented below.

The following diagram shows the series of events that occur when a timeline schedule page is accessed. Once the timeline has been loaded, all subsequent events, such as events resulting from timeline interaction (e.g. moving a timeline span), follow the same logic flow shown in the gray event box.



Client browser accesses a schedule page. The request is sent to the server.

```
http://instance.service-now.com/show_schedule_page.do?
sysparm_page_schedule_type=gantt_chart&sysparm_timeline_task_id=d530bf907f0000015ce594fd929cf6a4
```

The server interprets the schedule page HTTP request and obtains information about the specific schedule page from either of the following URI parameters:
**sysparm_page_sys_id**
or
**sysparm_page_schedule_type**

The server returns an HTTP response that contains the client script information from the specified schedule page.

The client browser immediately parses the **Client script** section of the schedule page upon loading of the page.

- Set configuration and display options.
- Register event listeners.

**Event: getItems**

The client timeline makes an AJAX request to the corresponding **Script Include** for the event registered with "getItems" to retrieve the set of items and spans to display on the timeline.

The server receives the request and executes the getItems code block inside the specified **Script Include** (which is an instance of the AbstractTimelineSchedulePage class). The server returns an XML document with TimelineItem objects.

The client receives the AJAX response with the specified TimelineItem objects and appropriately displays them on the screen.

## Applications that Use Schedule Pages to Generate Timelines

- **Project Management**
- **Maintenance Schedules**
- **Group On-Call Rotation**
- **Field Service Management**

# Application Programming Interface (API)

Timeline schedule pages use an internal class that allows for direct customization of how a timeline is displayed. Creation of new timeline schedule pages requires thorough understanding of how the page/event flow [link to above] works as well as capability to write both client and server side JavaScript. The API described below details the available methods and scripting requirements for both the client side and server side.

**Note:** ***Click here*** *to view a complete example of a new timeline schedule page with its corresponding script include.*

The timeline Application Programming Interface (API) contains the following classes:

- **Client Script / Class GlideTimeline**
- **Script Include / Class AbstractTimelineSchedulePage**
- **Class GlideTimelineItem**
- **Class TimelineSpan**

See the Incident Summary Timeline **example** for information about how a project support manager might use these APIs to create a timeline in which new incidents are grouped together by priority and closed incidents by duration.

## Client Script / Class GlideTimeline

The GlideTimeline class provides the core implementation for configuring and displaying a Glide Windowing Toolkit Timeline. For security the GlideTimeline has already been instantiated as a single instance variable named: **glideTimeline**. All configurations should be made in the client script section of the corresponding schedule page referencing this instance variable.

**Method Summary**

| Return Value | Details |
|---|---|
| void | **setReadOnly** `(Boolean b)`<br>Enables or disables all timeline event interaction. If enabled, event interaction is determined from the corresponding attributes specified by each Timeline Item. The default value for the readOnly property is `false`. |
| void | **showLeftPane** `(Boolean b)`<br>Specifies whether or not to show the left hand pane in the timeline. The default value for the showLeftPane property is `true`. |
| void | **showSummaryPane** `(Boolean b)`<br>Specifies whether or not to show the summary pane at the bottom of the timeline. The default value for the showSummaryPane property is `true`. |
| void | **showLeftPaneAsTree** `(Boolean b)`<br>Specifies whether or not to show items that have a parent attribute as nested indented nodes with expand and collapse capability. The default value for the showLeftPaneAsTree property is `false`. |

| | |
|---|---|
| void | **showTimelineText** `(Boolean b)`<br>Specifies whether or not to show the timeline text underneath each Timeline Span in the primary timeline pane. The default value for the showTimelineText property is `false`. |
| void | **showDependencyLines** `(Boolean b)`<br>Specifies whether or not to show dependency lines between Timeline Spans. This is only applicable if the set of Timeline Items returned from the server includes dependency relationships. The default value for the showDependencyLines property is `false`. |
| void | **groupByParent** `(Boolean b)`<br>Specifies whether or not to group items by their parent. If `true`, this will nest all child items inside their parent. This affects the ordering of display and children will always be listed immediately after their parent. The default value for the groupByParent property is `false`. |
| void | **sortByLeftLabelText** `(Boolean b)`<br>Specifies whether or not to sort the list of Timeline Items returned in alphabetical order according to the text that was specified to show in the Left Pane. Note that this function will still sort regardless if the left pane is disabled. Additionally, if groupByParent() is set `true`, items will be sorted appropriately after grouping has occurred. The default value for the sortByLeftLabelText property is `false`. |
| void | **sortByTimelineLabelText** `(Boolean b)`<br>Specifies whether or not to sort the list of Timeline Items returned in alphabetical order according to the text that was specified to show in the Timeline Pane. Note that this will still sort will still occur regardless if the timeline text has been set false via the showTimelineText() method. Additionally, if groupByParent() is set `true`, items will be sorted appropriately after grouping has occurred. The default value for the sortByTimelineLabelText property is `false`. |
| void | **sortByStartDate** `(Boolean b)`<br>Specifies whether or not to sort the list of Timeline Items returned by the earliest start date of an item's Timeline Span objects. Note that if groupByParent() is set `true`, items will be sorted appropriately after grouping has occurred. The default value for the sortByStartDate property is `false`. |
| void | **setDefaultPointIconClass** `(String strName)`<br>Specifies the default icon class to use for Timeline Spans with zero duration if no icon class was explicitly specified in the properties of the Timeline Span returned from the server. The default value for the setDefaultPointIconClass property is `milestone`. |
| void | **showLeftPaneInputBox** `(Boolean b, String strDefaultValue)`<br>Specifies whether or not to show the text input box at the bottom of the left pane with a default value as specified by `strDefaultValue`. Note that if the left pane is diabled via showLeftPane() the input box will not be visible. The default value for the showLeftPaneInputBox property is `false`. |
| void | **setInitialViewRange** `(mixed objStartDate, mixed objEndDate)`<br>Specifies the initial viewable range for the timeline. The format of the start and end date must be either in the default timestamp format: `yyyy-MM-dd HH:mm:ss` or specified as a number representing the time in milliseconds since the standard epoch of 1/1/1970. The default range is the range that specifies the earliest Timeline Span point to the end of the latest Timeline Span. If the initialViewRange property is specified, it will override the default range. |
| void | **setExtraAjaxParam** `(String strName, String strValue)`<br>Allows setting of additional parameters in the client script to be made available to the corresponding Script Include events by using the getParameter() method. URI parameters that are prefixed with "sysparm_timeline_" will automatically be included in all server side AJAX calls. |
| void | **registerEvent** `(String strServerEvent, String strScriptIncludeName)`<br>Registers the specified Timeline server event. The `strServerEvent` must be one of the allowed events for registration to work correctly. When the event occurs, the GlideTimeline will send a request to the server and process the event as handled inside the `strScriptIncludeName` class. |
| void | **snapVertScrollingIntoRows** `(Boolean boolSnapVertScrollingIntoRows)`<br>Specifies whether or not the vertical movement of timeline span objects (if appropriately registered to perform this event) should snap adjust into the closest row. By default this value is enabled. |

| | |
|---|---|
| void | **showGridLines** (Boolean boolShowGridLines)<br>Specifies whether or not to show grid lines for each row of data on the timeline. By default, grid lines are enabled. |
| void | **setAutoRefresh** (Number intSeconds)<br>Specifies the number of seconds to wait before performing an auto refresh of the data on the timeline. Setting the number of seconds to 0 will turn auto refresh off. By default, auto refresh is not enabled. Note that if `intSeconds` is greater than 0 and less than the minimum allowed time in seconds (10), it will be set to 10 seconds. |

Back to Application Programming Interface (API)

### setReadOnly

public void setReadOnly (Boolean b)

Enables or disables all timeline event interaction. If enabled, event interaction is determined from the corresponding attributes specified by each Timeline Item. The default value for the readOnly property is `false`.

**Parameters:**

`b` - Boolean variable that marks the entire timeline as read-only (non-interactive) if set to `true`.

**Example:**

```
glideTimeline.setReadOnly(true);
```

### showLeftPane

public void showLeftPane (Boolean b)

Specifies whether or not to show the left hand pane in the timeline. The default value for the leftPane property is `true`.

**Parameters:**

`b` - Boolean variable that will show the left pane if set `true`; otherwise, the left pane will not be displayed.

**Example:**

```
glideTimeline.showLeftPane(false);
```

### showSummaryPane

public void showSummaryPane (Boolean b)

Specifies whether or not to show the summary pane at the bottom of the timeline. The default value for the showSummaryPane property is `true`.

**Parameters:**

`b` - Boolean variable that will show the summary pane if set `true`; otherwise, if `false` the summary pane will not be displayed.

**Example:**

```
glideTimeline.showSummaryPane(false);
```

### showLeftPaneAsTree

public void showLeftPaneAsTree(Boolean b)

Specifies whether or not to show items that have a parent attribute as nested indented nodes with expand and collapse capability. The default value for the showLeftPaneAsTree property is false.

**Parameters:**

b - Boolean variable that will display child item nodes indented with expand/collapse capability if set true; otherwise, all left pane items will be displayed at a single indent level.

**Example:**

```
glideTimeline.showLeftPaneAsTree(true);
```

### showTimelineText

public void showTimelineText(Boolean b)

Specifies whether or not to show the timeline text underneath each Timeline Span in the primary timeline pane. The default value for the showTimelineText property is false.

**Parameters:**

b - Boolean variable that will display descriptive text underneath each Timeline Span if set true; otherwise, no text will be displayed underneath each Timeline Span.

**Example:**

```
glideTimeline.showTimelineText(true);
```

### showDependencyLines

public void showDependencyLines(Boolean b)

Specifies whether or not to show dependency lines between Timeline Spans. This is only applicable if the set of Timeline Items returned from the server includes dependency relationships. The default value for the showDependencyLines property is false.

**Parameters:**

b - Boolean variable that will display dependency lines on the timeline if set true; otherwise, will not.

**Example:**

```
glideTimeline.showDependencyLines(true);
```

### groupByParent

public void groupByParent(Boolean b)

Specifies whether or not to group items by their parent. If true, this will nest all child items inside their parent. This affects the ordering of display and children will always be listed immediately after their parent. The default value for the groupByParent property is false.

**Parameters:**

b - Boolean variable that will group Timeline Items by their parent if set true

**Example:**

```
glideTimeline.groupByParent(true);
```

### sortByLeftLabelText

public void sortByLeftLabelText`(Boolean b)`

Specifies whether or not to group items by their parent. If `true`, this will nest all child items inside their parent. This affects the ordering of display and children will always be listed immediately after their parent. The default value for the groupByParent property is `false`.

**Parameters:**

`b` - Boolean variable that will sort Timeline Items alphabetically by the text specified in an item's left label if set `true`

**Example:**

```
glideTimeline.sortByLeftLabelText(true);
```

### sortByTimelineLabelText

public void sortByTimelineLabelText`(Boolean b)`

Specifies whether or not to sort the list of Timeline Items returned in alphabetical order according to the text that was specified to show in the Timeline Pane. Note that this will still sort will still occur regardless if the timeline text has been set false via the showTimelineText() method. Additionally, if groupByParent() is set `true`, items will be sorted appropriately after grouping has occurred. The default value for the sortByTimelineLabelText property is `false`.

**Parameters:**

`b` - Boolean variable that will sort Timeline Items alphabetically by the text specified in an item's timeline span text if set `true`

**Example:**

```
glideTimeline.sortByTimelineLabelText(true);
```

### sortByStartDate

public void sortByStartDate`(Boolean b)`

Specifies whether or not to sort the list of Timeline Items returned by the earliest start date of an item's Timeline Span objects. Note that if groupByParent() is set `true`, items will be sorted appropriately after grouping has occurred. The default value for the sortByStartDate property is `false`.

**Parameters:**

`b` - Boolean variable that will sort Timeline Items chronologically to their earliest start date if set `true`

**Example:**

```
glideTimeline.sortByStartDate(true);
```

## setDefaultPointIconClass

public void sortByStartDate (`String strName`)

Specifies the default icon class to use for Timeline Spans with zero duration if no icon class was explicitly specified in the properties of the Timeline Span returned from the server. The default value for the setDefaultPointIconClass property is `milestone`.

**Parameters:**

`strName` - String that specifies one of the following values:

- `milestone` ◆
- `blue_square` 🟦
- `sepia_square` 🟫
- `green_square` 🟩
- `red_square` 🟥
- `black_square` ⬛
- `blue_circle` 🔵
- `sepia_circle` 🟤
- `green_circle` 🟢
- `red_circle` 🔴
- `black_circle` ⚫

**Example:**

```
glideTimeline.setDefaultPointIconClass('blue_circle');
```

## showLeftPaneInputBox

public void showLeftPaneInputBox (`Boolean b, String strDefaultValue`)

Specifies whether or not to show the text input box at the bottom of the left pane with a default value as specified by `strDefaultValue`. Note that if the left pane is diabled via showLeftPane() the input box will not be visible. The default value for the showLeftPaneInputBox property is `false`.

**Parameters:**

`b` - Boolean variable that will show the left pane input box if set `true`.

`strDefaultValue` - String that specifies the default value to display in the input box.

**Example:**

```
glideTimeline.showLeftPaneInputBox(true, 'Add a new task ...');
```

## setInitialViewRange

public void setInitialViewRange (`mixed objStartDate, mixed objEndDate`)

Specifies the initial viewable range for the timeline. The format of the start and end date must be in the default timestamp format: `yyyy-MM-dd HH:mm:ss`. The default range is the range that specifies the earliest Timeline Span point to the end of the latest Timeline Span. If the initialViewRange property is specified, it will override the default range.

**Parameters:**

`objStartDate` - Either a string that specifies the start time of the view range in format: `yyyy-MM-dd HH:mm:ss` or a number representing the start time in milliseconds.

`objEndDate` - Either a string that specifies the end time of the view range in format: `yyyy-MM-dd HH:mm:ss` or a number representing the end time in milliseconds.

**Example:**

```
// Sets the initial range to begin on June 20th, 2010 at 8:00 AM and
end on June 28th, 2010 at 2:00 PM UTC time.
glideTimeline.setInitialViewRange("2010-06-20 08:00:00",
1277647200000);
```

## setExtraAjaxParam

public void setExtraAjaxParam `(String strStartDate, String strEndDate)`

Allows setting of additional parameters in the client script to be made available to the corresponding Script Include events by using the getParameter() method. URI parameters that are prefixed with "sysparm_timeline_" will automatically be included in all server side AJAX calls..

**Parameters:**

`strName` - String that specifies the URI parameter name.

`strValue` - String that specifies the value of `strName`.

**Example:**

```
glideTimeline.setExtraAjaxParam("sysparm_timeline_limit", "5");
```

## registerEvent

public void registerEvent `(String strServerEvent, String strScriptIncludeName)`

Registers the specified Timeline server event. The `strServerEvent` must be one of the allowed events for registration to work correctly. When the event occurs, the GlideTimeline will send a request to the server and process the event as handled inside the `strScriptIncludeName` class.

**Parameters:**

`strServerEvent` - String that specifies one of the following **case-sensitive** events:

- `getItems`
- `elementMoveX`
- `elementMoveY`
- `elementMoveXY`
- `elementSuccessor`
- `elementTimeAdjustStart`
- `elementTimeAdjustEnd`
- `inputBox`
- `itemMove`

`strScriptIncludeName` - String that specifies the name of the class to receive the `strServerEvent`. This class should must be defined in a script include that extends `AbstractTimelineSchedulePage`.

**Example:**

```
glideTimeline.registerEvent("getItems", "TimelineGanttSchedulePage");
```

### snapVertScrollingIntoRows

public void snapVertScrollingIntoRows `(Boolean b)`

Specifies whether or not the vertical movement of timeline span objects (if appropriately registered to perform this event) should snap adjust into the closest row. By default this value is enabled.

**Parameters:**

`b` - Boolean variable that will snap vertical movement into rows if set `true`; otherwise, items will move exactly with respect to the mouse.

**Example:**

```
glideTimeline.snapVertScrollingIntoRows(false);
```

### showGridLines

public void showGridLines `(Boolean boolShowGridLines)`

Specifies whether or not to show grid lines for each row of data on the timeline. By default, grid lines are enabled.

**Parameters:**

`boolShowGridLines` - Boolean variable that will show grid lines if set `true`; otherwise, grid lines will not be displayed.

**Example:**

```
glideTimeline.showGridLines(false); // Disables grid lines.
```

### setAutoRefresh

public void setAutoRefresh `(Number intSeconds)`

Specifies the number of seconds to wait before performing an auto refresh of the data on the timeline. Setting the number of seconds to 0 will turn auto refresh off. By default, auto refresh is disabled. Note that if `intSeconds` is greater than 0 and less than the minimum allowed time in seconds (10), it will be set to 10 seconds.

**Parameters:**

`intSeconds` - Integer that specifies time in seconds between auto-refreshing.

**Example:**

```
glideTimeline.setAutoRefresh(15); // Sets the interval for
auto-refreshing to 15 seconds.
```

## Script Include / Class AbstractTimelineSchedulePage

The processing of all data displayed within a timeline is performed first by utilizing the corresponding function of the specified script include. Like other script includes, the language syntax is JavaScript and follows the default security constraints of this type of resource. However, in order to deal with the complexity of the different types of display options a helper class was created. The `AbstractTimelineSchedulePage` is an abstract class that must be extended and paired with it's corresponding schedule page to display a timeline. At a minimum, extending classes should override the `getItems()` method as this is the primary event handler for returning items to be displayed on the client.

The AbstractTimelineSchedulePage returns data to the client that is processed in two phases. The first phase processes the data making actual updates to the timeline. Immediately after (if configured) the second phase will display a success message box, error message box, or dialog message prompt. The available options in phase one for manipulating data include:

1. **Do Not Update Any Items** - This is the default behavior. Do not perform any of the remaining steps in phase one.
2. **Update With Specific Items** - This is done using: `add()`.
3. **Render The Timeline Using the getItems() Function** - This is done using: `setDoReRenderTimeline(true)`.

**Note:** *If both TimelineItems are returned and `setDoReRenderTimeline` is set to true, the system will ignore the `setDoReRenderTimeline` property and explicitly show only the TimelineItems that were added via the `add()` function.*

The available options in phase two, which control optional message boxes after phase one is complete include:

1. **Do Not Display Any Message Boxes** - This is the default behavior.
2. **Display a Success Dialog Box** - This is done using: `setStatusSuccess()`.
3. **Display an Error Dialog Box** - This is done using: `setStatusError()`.
4. **Display a Dialog Confirm Box** - This is done using: `setStatusPrompt()`.

For each dialog function, see the corresponding API below for proper implementation.

**Note:** *A script include class that extends `AbstractTimelineSchedulePage` will automatically receive all Uri parameters from the original Url whose prefix begins with "**sysparm_timeline_**". To access the values of these, use: `this.getParameter("sysparm_timeline_VARIABLE");` inside your extended class. This is useful if you need to display a schedule page from a dynamic element, such as from a context menu from a list. By passing in dynamic data via the Url the schedule page will auto-include these parameters inside the Ajax calls and therefore will be accessible inside the AbstractTimelineSchedulePage script includes.*

## Method Summary

The following methods exist to facilitate data manipulation when extending the event handling methods in the following section.

| Return Value | Details |
|---|---|
| void | **add**`(TimelineItem objItem)`<br>Adds a TimelineItem object that will be returned to the client and appropriately displayed on the timeline. |
| void | **addSeparator**`()`<br>Adds a horizontal frame separator into the list of timeline items. All future items added via `add()` will be added into the subsequent timeline frame. |
| void | **setPageTitle**`(String strTitle)`<br>Specifies the text to display as the title of the timeline. The page title can be set (and updated) from any interactive event; however, is recommended to be set during the `getItems()` event. |
| void | **setDoReRenderTimeline**`(Boolean b)`<br>Specifies whether or not to re-render all of the timeline items using the `getItems()` function. |
| void | **setStatusError**`(String strTitle, String strMessage)`<br>Denotes the current event request with an error status that will display a dialog box with the specified title and message immediately in phase two of the GlideTimeline event processing. |
| void | **setStatusSuccess**`(String strTitle, String strMessage)`<br>Denotes the current event request with a success status that will display a dialog box with the specified title and message immediately in phase two of the GlideTimeline event processing. |

| | |
|---|---|
| void | **setStatusPrompt** `(String strTitle, String strMessage, String strOkFunction, String strCancelFunction, String strCloseFunction)`<br>Denotes the current event request with a prompt status that will display a confirmation dialog box with the specified title and message immediately in phase two of the GlideTimeline event processing. The confirmation box displays an "OK" and "Cancel" button that each generate new events that will call the specified functions as denoted in the parameter arguments. **Note that the custom defined functions for the OK, Cancel, and Close will receive the same parameter arguments as those for the current event.** |

## Abstract Event Handler Methods

The following methods correspond with the specified event thrown from the schedule page.

| Return Value | Details |
|---|---|
| void | **getItems** `()`<br>Event handler for returning schedule items to display on the timeline. |
| void | **elementMoveX** `(String spanSysId, String newStartDateTimeMs)`<br>Event handler for when a timeline span moves horizontally. |
| void | **elementMoveY** `(String spanSysId, String itemSysId, String newItemSysId)`<br>Event handler for when a timeline span moves vertically. |
| void | **elementMoveXY** `(String spanSysId, String itemSysId, String newItemSysId, String newStartDateTimeMs)`<br>Event handler for when a timeline span moves both horizontally and vertically. |
| void | **elementSuccessor** `(sstring spanSysId, String newSuccSpanId)`<br>Event handler for when a timeline relationship has been created between two spans. |
| void | **elementTimeAdjustStart** `(String spanSysId, String newStartDateTimeMs)`<br>Event handler for when a timeline span's start date was modified. |
| void | **elementTimeAdjustEnd** `(String spanSysId, String newStartDateTimeMs)`<br>Event handler for when a timeline span's end date was modified. |
| void | **inputBox** `(String strInputText)`<br>Event handler for when a string was typed into the left pane input box. |
| void | **itemMove** `(String itemSysId, String newItemSysId)`<br>Event handler for when a timeline row item was moved and dragged into another row item. |

Back to Application Programming Interface (API)

## add

public void add `(GlideTimelineItem objItem)`

Adds a GlideTimelineItem object that will be returned to the client and appropriately displayed on the timeline.

**Parameters:**

`objItem` - The GlideTimelineItem object to add to the timeline.

## addSeparator

public void addSeparator `()`

Adds a horizontal frame separator into the list of timeline items. All future items added via `add()` will be added into the subsequent timeline frame.

**Example:**

```
// Inside of a script include that extends AbstractTimelineSchedulePage
this.addSeparator();
```

## setPageTitle

public void setPageTitle (`String strTitle`)

Specifies the text to display as the title of the timeline. The page title can be set (and updated) from any interactive event; however, is recommended to be set during the `getItems()` event.

**Parameters:**

`strTitle` - String that specifies the text to be displayed on the title of the timeline.

## setDoReRenderTimeline

public void setDoReRenderTimeline (`Boolean b`)

Specifies whether or not to re-render all of the timeline items using the `getItems()` function.

**Parameters:**

`b` - Boolean variable that will re-render the timeline by making a new event call to the server's `getItems()` handler if set `true`.

## setStatusError

public void setStatusError (`String strTitle, String strMessage`)

Denotes the current event request with an error status that will display a dialog box with the specified title and message immediately in phase two of the GlideTimeline event processing.

**Parameters:**

`strTitle` - String that specifies the text to be displayed in the dialog box title.

`strMessage` - String that specifies the text to be displayed within the dialog box. The text can contain HTML formatting.

## setStatusSuccess

public void setStatusSuccess (`String strTitle, String strMessage`)

Denotes the current event request with a success status that will display a dialog box with the specified title and message immediately in phase two of the GlideTimeline event processing.

**Parameters:**

`strTitle` - String that specifies the text to be displayed in the dialog box title.

`strMessage` - String that specifies the text to be displayed within the dialog box. The text can contain HTML formatting.

### setStatusPrompt

public void setStatusPrompt(`String strTitle, String strMessage, String strOkFunction, String strCancelFunction, String strCloseFunction`)

Denotes the current event request with a prompt status that will display a confirmation dialog box with the specified title and message immediately in phase two of the GlideTimeline event processing. The confirmation box displays an "OK" and "Cancel" button that each generate new events that will call the specified functions as denoted in the parameter arguments. **Note that the custom defined functions for the OK, Cancel, and Close will receive the same parameter arguments as those for the current event.**

**Parameters:**

`strTitle` - String that specifies the text to be displayed in the dialog box title.

`strMessage` - String that specifies the text to be displayed within the dialog box. The text can contain HTML formatting.

`strOkFunction` - String that specifies the name of the function to call in the current script include class if the "OK" button is clicked.

`strCancelFunction` - String that specifies the name of the function to call in the current script include class if the "Cancel" button is clicked.

`strCloseFunction` - String that specifies the name of the function to call in the current script include class if the "Close" button is clicked.

**Example:**

```
var MyTimelineScriptIncludeClass = Class.create();
MyTimelineScriptIncludeClass.prototype =
Object.extendsObject(AbstractTimelineSchedulePage, {

  getItems: function() {
    //...
  },

  elementTimeAdjustEnd: function(spanSysId, newEndDateTimeMs) {
    // Display a status prompt dialog box
    this.setStatusPrompt('Confirm Action', 'Are you sure you want to do
 that?',
                         'this._myOkHandlerFunction',
                         'this._myCancelHandlerFunction',
                         'this._myCloseHandlerFunction');
  },

  _myOkHandlerFunction: function(spanSysId, newEndDateTimeMs) { // ...
},

  _myCancelHandlerFunction: function(spanSysId, newEndDateTimeMs) { //
... },

  _myCloseHandlerFunction: function(spanSysId, newEndDateTimeMs) { //
... }
};
```

### getItems

public void getItems()

Event handler for returning schedule items to display on the timeline.

### elementMoveX

public void elementMoveX `(String spanSysId, String newStartDateTimeMs)`

Event handler for when a timeline span moves horizontally.

**Parameters:**

`spanSysId` - String that specifies the sys ID of the current span that is being adjusted.

`newStartDateTimeMs` - String that specifies the new start time of the span in milliseconds. Make sure to parse the string using `parseInt()` prior to performing any numerical comparisons.

### elementMoveY

public void elementMoveY `(String spanSysId, String itemSysId, String newItemSysId)`

Event handler for when a timeline span moves vertically.

**Parameters:**

`spanSysId` - String that specifies the sys ID of the current span that is being adjusted.

`itemSysId` - String that specifies the sys ID of the timeline item associated with the current span.

`newItemSysId` - String that specifies the sys ID of the timeline item (a row) that the current span was dragged into.

### elementMoveXY

public void elementMoveXY `(String spanSysId, String itemSysId, String newItemSysId, String newStartDateTimeMs)`

Event handler for when a timeline span moves both horizontally and vertically.

**Parameters:**

`spanSysId` - String that specifies the sys ID of the current span that is being adjusted.

`itemSysId` - String that specifies the sys ID of the timeline item associated with the current span.

`newItemSysId` - String that specifies the sys ID of the timeline item (a row) that the current span was dragged into.

`newStartDateTimeMs` - String that specifies the new start time of the span in milliseconds. Make sure to parse the string using `parseInt()` prior to performing any numerical comparisons.

### elementSuccessor

public void elementSuccessor `(String spanSysId, String newSuccSpanId)`

Event handler for when a timeline relationship has been created between two spans.

**Parameters:**

`spanSysId` - String that specifies the sys ID of the current span which will be a predecessor for the newly created relationship.

`newSuccSpanId` - String that specifies the sys ID of the successor span from the relationship created.

### elementTimeAdjustStart

public void elementTimeAdjustStart `(String spanSysId, String newStartDateTimeMs)`

Event handler for when a timeline span's start date was modified.

**Parameters:**

`spanSysId` - String that specifies the sys ID of the current span that is being adjusted.

`newStartDateTimeMs` - String that specifies the new start time of the span in milliseconds. Make sure to parse the string using `parseInt()` prior to performing any numerical comparisons.

### elementTimeAdjustEnd

public void elementTimeAdjustEnd `(String spanSysId, String newStartDateTimeMs)`

Event handler for when a timeline span's end date was modified.

**Parameters:**

`spanSysId` - String that specifies the sys ID of the current span that is being adjusted.

`newStartDateTimeMs` - String that specifies the new start time of the span in milliseconds. Make sure to parse the string using `parseInt()` prior to performing any numerical comparisons.

### inputBox

public void inputBox `(String strInputText)`

Event handler for when a string was typed into the left pane input box.

**Parameters:**

`strInputText` - String that specifies the text that was entered in the input box in the left pane

### itemMove

public void itemMove `(String itemSysId, String newItemSysId)`

Event handler for when a timeline row item was moved and dragged into another row item.

**Parameters:**

`itemSysId` - String that specifies the sys ID of the timeline item associated with the current span.

`newItemSysId` - String that specifies the sys ID of the timeline item (a row) that the current span was dragged into.

## Class GlideTimelineItem

This class extends the abstract `ScheduleItem` class to define additional properties that are specific to the timeline. A timeline item is essentially any item that is displayed in a singular row across the timeline. A `GlideTimelineItem` may exist where it has zero or more spans `(TimelineSpan` objects) associated with it.

**See Also:**

**TimelineSpan**

## Constructor Summary

| Return Value | Details |
|---|---|
| GlideTimelineItem | **GlideTimelineItem** `(String sys_id)`<br>Non-default constructor that allows a "dummy" GlideTimelineItem to be created. Useful for creating rows that do not allow any YMoving into; however, contain nested children. (e.g. The top-level "Users" row in the Group Resource Timeline). The sys_id needs to be unique for DOM level functions to parse correctly. |
| GlideTimelineItem | **GlideTimelineItem** `(String table, String sys_id)`<br>Constructor that sets the required table and sys_id properties. The rest of this object's properties should be set from the caller appropriately. |

## Method Summary

| Return Value | Details |
|---|---|
| TimelineSpan | **createTimelineSpan** `(String tableName)`<br>Creates a new `TimelineSpan` object associated with the current instance object. If no other `TimelineSpan` objects exist, the newly created object will share the same sys_id as current instance object. Otherwise, a randomly generated GUID will be used. |
| TimelineSpan | **createTimelineSpan** `(String tableName, String sysId)`<br>Creates a new `TimelineSpan` object associated with the current instance object using the specified table and sysId. |
| String | **getImage** `()`<br>Returns a string specifying the name of the image file associated with the current image. If no image is associated with the current image, an empty string ("") is returned. |
| Boolean | **getIsDroppable** `()`<br>Returns a boolean that specifies whether or not the current instance object should be allowed as a "drop zone" when moving timeline elements vertically. |
| String | **getLeftLabelText** `()`<br>Returns a string that specifies the text to display in the left pane (if enabled). |
| String | **getParent** `()`<br>Returns the unique sysId of the current `TimelineItem`'s parent object. If the parent does not exist, this will return an empty string (""). |
| ArrayList | **getTimelineSpans** `()`<br>Returns all the `TimelineSpan` objects associated with the current instance in an ArrayList. |
| Boolean | **isTextBold** `()`<br>Returns a boolean that specifies if the left pane text should be displayed using a **bold** style. |
| void | **setImage** `(String strImageName)`<br>Specifies the name of the image file (including it's path) to use as the icon for the item in the left pane. |
| void | **setIsDraggable** `(Boolean b)`<br>Specifies whether or not the current instance object can be clicked and dragged into another Timeline Item. |
| void | **setIsDroppable** `(Boolean b)`<br>Specifies whether or not the current instance object should be allowed as a "drop zone" when moving timeline elements vertically. |
| void | **setLeftLabelText** `(String strText)`<br>Specifies the text to display in the left pane for this item. |
| void | **setParent** `(String parentSysId)`<br>Specifies the sys ID of another `TimelineItem` who is a parent of the current instance. |

| | |
|---|---|
| void | **setTextBold**`(Boolean b)`<br>Specifies whether or not to bold the text style of the item in the left pane. |

Back to Application Programming Interface (API)

## GlideTimelineItem

public GlideTimelineItem `(String tableName)`

Non-default constructor that allows a "dummy" GlideTimelineItem to be created. Useful for creating rows that do not allow any YMoving into; however, contain nested children. (e.g. The top-level "Users" row in the Group Resource Timeline). The sys_id needs to be unique for DOM level functions to parse correctly. By default this object will **not** be "droppable" because a table name was not specified.

**Parameters:**

`tableName` - String that specifies the name of the table that is associated with current object.

## GlideTimelineItem

public GlideTimelineItem `(String tableName, String sys_id)`

Constructor that sets the required table and sys_id properties. The rest of this object's properties should be set from the caller appropriately. By default, this object instance is "droppable" since a table name was specified.

**Parameters:**

`tableName` - String that specifies the name of the table that is associated with current object.

`sys_id` - String that specifies the sys ID for the object.

## createTimelineSpan

public TimelineSpan createTimelineSpan `(String tableName)`

Creates a new `TimelineSpan` object associated with the current instance object. If no other `TimelineSpan` objects exist, the newly created object will share the same sys_id as current instance object. Otherwise, a randomly generated GUID will be used.

**Parameters:**

`tableName` - String that specifies the name of the table that is associated with current object.

**Returns:**

`TimelineSpan` - The newly created span object instance.

## createTimelineSpan

public TimelineSpan createTimelineSpan `(String tableName, String sysId)`

Creates a new TimelineSpan object associated with the current instance object using the specified table and sysId.

**Parameters:**

`tableName` - String that specifies the name of the table that is associated with current object.

`sys_id` - String that specifies the sys ID for the object.

**Returns:**

`TimelineSpan` - The newly created span object instance.

### getImage

public String getImage()

Returns a string specifying the name of the image file associated with the current image. If no image is associated with the current image, an empty string ("") is returned.

**Returns:**

`String` - The String name of the image file and path if specified; otherwise "".

### getIsDroppable

public Boolean getIsDroppable()

Returns a boolean that specifies whether or not the current instance object should be allowed as a "drop zone" when moving timeline elements vertically.

**Returns:**

`Boolean` - `true` if droppable; `false` otherwise.

### getLeftLabelText

public String getLeftLabelText()

Returns a string that specifies the text to display in the left pane (if enabled).

**Returns:**

`String` - The String value of the left label text if specified; otherwise "".

### getParent

public String getParent()

Returns the unique sysId of the current TimelineItem's parent object. If the parent does not exist, this will return an empty string ("").

**Returns:**

`String` - The String that specifies the parent `TimelineItem` of the current instance if specified; otherwise "".

### getTimelineSpans

public ArrayList getTimelineSpans()

Returns all the TimelineSpan objects associated with the current instance in an ArrayList.

**Returns:**

`ArrayList` - The list of `TimelineSpan` objects associated with the current instance.

### isTextBold

public Boolean isTextBold()

Returns a boolean that specifies if the left pane text should be displayed using a bold style.

**Returns:**

`Boolean` - `true` if the text should be **bolded**; otherwise `false`.

### setImage

public void setImage (`String strImageName`)

Specifies the name of the image file (including it's path) to use as the icon for the item in the left pane.

**Parameters:**

`strImageName` - The name of the image including it's path.

### setIsDraggable

public void setIsDraggable (`Boolean b`)

Specifies whether or not the current instance object can be clicked and dragged into another Timeline Item.

**Parameters:**

`b` - `true` if the item can be clicked and moved; otherwise, `false`.

### setIsDroppable

public void setIsDroppable (`Boolean b`)

Specifies the name of the image file (including it's path) to use as the icon for the item in the left pane.

**Parameters:**

`b` - `true` if the item should be marked as "droppable"; otherwise, `false`.

### setLeftLabelText

public void setLeftLabelText (`String strText`)

Specifies the text to display in the left pane for this item.

**Parameters:**

`strText` - The text to display in the left pane for this item.

### setParent

public void setParent (`String parentSysId`)

Specifies the sys ID of another `TimelineItem` who is a parent of the current instance.

**Parameters:**

`parentSysId` - Parent `TimelineItem` sys Id.

### setTextBold

public void setTextBold (`Boolean b`)

Specifies whether or not to bold the text style of the item in the left pane.

**Parameters:**

`b` - `true` if the style of the left pane item text should be **bold**; otherwise `false`.

## Class TimelineSpan

This class defines a set of properties that describe the characteristics and interactive behavior of an element rendered within a `GlideTimelineItem`. Since it is important for all of a `GlideTimelineItem`'s collection of spans to be unique, the creation of a new instance should be performed via the `createTimelineItem` method of an existing `GlideTimelineItem` instance.

**See Also:**

**GlideTimelineItem**

| Method Summary | |
|---|---|
| **Return Value** | **Details** |
| void | **addPredecessor**`(Object[] objArray)`<br>Adds multiple relationships between the current instance and other `TimelineSpan` objects by enumerating through the array of JavaScript objects. Each object should have an internal property "relationship_sys_id" and "predecessor_sys_id" specified. |
| void | **addPredecessor**`(String strPredecessorSysId, String strRelationshipSysId)`<br>Adds the specified relationship between the current instance and another `TimelineSpan` with sys ID `strPredecessorSysId`. The drawn line will not have any double click handlers associated with it. |
| void | **addPredecessor**`(String strPredecessorSysId, String strRelationshipSysId, String strTableName)`<br>Adds the specified relationship between the current instance and another `TimelineSpan` with sys ID `strPredecessorSysId` and allow the relationship to open a GlideWindow to display information about the relationship. |
| Boolean | **getAllowXDragLeft**`()`<br>Returns the boolean value of the AllowXDragLeft property. |
| Boolean | **getAllowXDragRight**`()`<br>Returns the boolean value of the AllowXDragRight property. |
| Boolean | **getAllowXMove**`()`<br>Returns the boolean value of the AllowXMove property. |
| Boolean | **getAllowYMove**`()`<br>Returns the boolean value of the AllowYMove property. |
| Boolean | **getAllowYMovePredecessor**`()`<br>Returns the boolean value of the AllowYMovePredecessor property. |
| Number | **getEndTimeMs**`()`<br>Returns the start time in milliseconds of the current instance object as a long Number. |
| String | **getInnerSegmentClass**`()`<br>Returns a String that specifies the current inner segment class for the Timeline Span. |
| Number | **getInnerSegmentEndTimeMs**`()`<br>Returns the time in milliseconds of the end time of the inner segment portion of the timeline span as a long Number. |
| Number | **getInnerSegmentStartTimeMs**`()`<br>Returns the time in milliseconds of the start time of the inner segment portion of the timeline span as a long Number. |
| Boolean | **getIsChanged**`()`<br>Returns a boolean that specifies whether or not the current timeline item has been modified after initialization. |
| String | **getPointIconClass**`()`<br>Returns a String that specifies the name of the icon class to use for displaying the element on the timeline if the current instance has zero duration. |

| | |
|---|---|
| ArrayList | **getPredecessors**`()`<br>Returns an ArrayList of all the predecessor objects associated with the current instance. Each ArrayList object is a HashMap that contains a `predecessor_sys_id` and `relationship_sys_id` property. |
| String | **getSpanColor**`()`<br>Returns the string name of the color specified for displaying this span. |
| String | **getSpanText**`()`<br>Returns the string that specifies the text to display adjacent to the time element. Note that this text will only be displayed if the GlideTimeline object has enabled timeline text via `glideTimeline.showTimelineText(true).` |
| Number | **getStartTimeMs**`()`<br>Returns the start time in milliseconds of the current instance object as a long Number. |
| String | **getSysId**`()`<br>Returns the sys ID of the current object. This method is useful for returning the sys Id when the current object instance was created without a specific sys Id to obtain the dynamically generated GUID. |
| String | **getTable**`()`<br>Returns the string name of the table where the sys ID is referenced. |
| String | **getTooltip**`()`<br>Returns the string that specifies the text/html to display in the tooltip when the TimeSpan element is being hovered over. |
| void | **setAllowXDragLeft**`(Boolean b)`<br>Sets a flag that determines whether the element's start date can be dragged left or right therefore adjusting the duration of the task. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is `false`. |
| void | **setAllowXDragRight**`(Boolean b)`<br>Sets a flag that determines whether the element's end date can be dragged left or right therefore adjusting the duration of the task. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is `false`. |
| void | **setAllowXMove**`(Boolean b)`<br>Sets a flag that determines whether the element can be moved to start at a different time. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is `false`. |
| void | **setAllowYMove**`(Boolean b)`<br>Sets a flag that determines whether the element can be dragged vertically on the timeline. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is `false`. |
| void | **setAllowYMovePredecessor**`(Boolean b)`<br>Sets a flag that determines whether a dashed relationship line can be drawn from this element interactively on the timeline. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is `false`. |
| void | **setInnerSegmentClass**`(String s)`<br>Specifies the name of the class to use for stylizing the inner segment if it exists. |
| void | **setInnerSegmentTimeSpan**`(String startTimeMs, String endTimeMs)`<br>Creates an inner segment to show within the current timespan defined by the range specified. |
| void | **setInnerSegmentTimeSpan**`(Number startTimeMs, Number endTimeMs)`<br>Creates an inner segment to show within the current timespan defined by the range specified. |
| void | **setIsChanged**`(Boolean b)`<br>Sets a flag that specifies whether or not the current timeline item has been modified after initialization. The default value for this property is `false`. |

| | |
|---|---|
| void | **setPointIconClass** `(String s)`<br>Sets the icon class to use for displaying the current element on the timeline if the current instance has zero duration. Note that this only affects the current `TimelineSpan` object and will take precedence over the defaultPointIconClass specified by the `GlideTimeline`. |
| void | **setSpanColor** `(String s)`<br>Sets the color for displaying this span. This needs to be a valid HTML color entity specified by a named color or a hexadecimal color code. |
| void | **setSpanText** `(String s)`<br>Sets the text to display adjacent to the time element. Note that this text will only be displayed if the GlideTimeline object has enabled timeline text via `glideTimeline.showTimelineText(true)`. |
| void | **setTimeSpan** `(long startTimeMs, long endTimeMs)`<br>Sets the start and end dates for the current span. |
| void | **setTimeSpan** `(String startTimeMs, String endTimeMs)`<br>Sets the start and end dates for the current span. |
| void | **setTooltip** `(String s)`<br>Sets the text to display in the tooltip when the TimeSpan element is being hovered over.<br>**Note:** *You can specify valid HTML in the string that sets the tooltip.* |

Back to Application Programming Interface (API)

## addPredecessor

public void addPredecessor `(Object[] objArray)`

Adds multiple relationships between the current instance and other `TimelineSpan` objects by enumerating through the array of JavaScript objects. Each object should have an internal property "**relationship_sys_id**" and "**predecessor_sys_id**" specified.

**Parameters:**

`objArray` - JavaScript object array that contains two internal properties: **relationship_sys_id** and **predecessor_sys_id**.

## addPredecessor

public void addPredecessor `(String strPredecessorSysId, String strRelationshipSysId)`

Adds the specified relationship between the current instance and another `TimelineSpan` with sys ID `strPredecessorSysId`. The drawn line will not have any double click handlers associated with it.

**Parameters:**

`strPredecessorSysId` - String that specifies the sys ID of the planned task that is the predecessor of the relationship.

`strRelationshipSysId` - String that specifies the sys ID of the relationship of the relationship.

### addPredecessor

public void addPredecessor`(String strPredecessorSysId, String strRelationshipSysId, String strTableName)`

Adds the specified relationship between the current instance and another `TimelineSpan` with sys ID `strPredecessorSysId` and allow the relationship to open a GlideWindow to display information about the relationship.

**Parameters:**

`strPredecessorSysId` - String that specifies the sys ID of the planned task that is the predecessor of the relationship.

`strRelationshipSysId` - String that specifies the sys ID of the relationship of the relationship.

`strTableName` - String that specifies the name of the table for the relationship.

### getAllowXDragLeft

public Boolean getAllowXDragLeft`()`

Returns the boolean value of the AllowXDragLeft property.

**Returns:**

`Boolean` - `true` if the object's start time can be adjusted; `false` otherwise.

### getAllowXDragRight

public Boolean getAllowXDragRight`()`

Returns the boolean value of the AllowXDragRight property.

**Returns:**

`Boolean` - `true` if the object's end time can be adjusted; `false` otherwise.

### getAllowXMove

public Boolean getAllowXMove`()`

Returns the boolean value of the AllowXMove property.

**Returns:**

`Boolean` - `true` if the object can be moved; `false` otherwise.

### getAllowYMove

public Boolean getAllowYMove`()`

Returns the boolean value of the AllowYMove property.

**Returns:**

`Boolean` - `true` if the object can be moved vertically; `false` otherwise.

### getAllowYMovePredecessor

public Boolean getAllowYMovePredecessor ()

Returns the boolean value of the AllowYMovePredecessor property.

**Returns:**

`Boolean` - `true` if the a dashed relationship line can be drawn from the current object to a new successor; `false` otherwise.

### getEndTimeMs

public Number getEndTimeMs ()

Returns the start time in milliseconds of the current instance object as a long Number.

**Returns:**

`Number` - The end time of the timeline span in milliseconds.

### getInnerSegmentClass

public String getInnerSegmentClass ()

Returns a String that specifies the current inner segment class for the Timeline Span.

**Returns:**

`String` - The String name of the class for the current inner segment style.

### getInnerSegmentEndTimeMs

public Number getInnerSegmentEndTimeMs ()

Returns the time in milliseconds of the end time of the inner segment portion of the timeline span as a long Number.

**Returns:**

`Number` - The end time of the timeline span inner segment portion in milliseconds.

### getInnerSegmentStartTimeMs

public Number getInnerSegmentStartTimeMs ()

Returns the time in milliseconds of the start time of the inner segment portion of the timeline span as a long Number.

**Returns:**

`Number` - The start time of the timeline span inner segment portion in milliseconds.

### getIsChanged

public Boolean getIsChanged`()`

Returns a boolean that specifies whether or not the current timeline item has been modified after initialization.

**Returns:**

`Boolean` - `true` if the current span has been marked as **changed**; otherwise `false`.

### getPointIconClass

public String getPointIconClass`()`

Returns a String that specifies the name of the icon class to use for displaying the element on the timeline if the current instance has zero duration.

**Returns:**

`String` - The name of the icon class to use for displaying the current `TimelineSpan` if the duration is zero.

### getPredecessors

public ArrayList getPredecessors`()`

Returns an ArrayList of all the predecessor objects associated with the current instance. Each ArrayList object is a HashMap that contains a **predecessor_sys_id** and **relationship_sys_id** property.

**Returns:**

`ArrayList` - List of HashMap's that contain two internal properties: **predecessor_sys_id** and **relationship_sys_id**.

### getSpanColor

public String getSpanColor`()`

Returns the string name of the color specified for displaying this span.

**Returns:**

`String` - String that specifies the HTML color name to use as the background color for the element.

### getSpanText

public String getSpanText`()`

Returns the string that specifies the text to display adjacent to the time element.

**Note:** *Note that this text will only be displayed if the GlideTimeline object has enabled timeline text via* `glideTimeline.showTimelineText(true)`*.*

**Returns:**

`String` - String value that contains the text to display adjacent to the element.

### getStartTimeMs

public Number getStartTimeMs ()

Returns the start time in milliseconds of the current instance object as a long Number.

**Returns:**

`Number` - String that specifies the start time of the element in miliseconds.

### getSysId

public String getSysId ()

Returns the sys ID of the current object. This method is useful for returning the sys Id when the current object instance was created without a specific sys Id to obtain the dynamically generated GUID.

**Returns:**

`String` - String that specifies the unique system ID of the current element.

### getTable

public String getTable ()

Returns the string name of the table where the sys ID is referenced.

**Returns:**

`String` - String that specifies the table name.

### getTooltip

public String getTooltip ()

Returns the string that specifies the text/html to display in the tooltip when the TimeSpan element is being hovered over.

**Returns:**

`String` - String that specifies the tooltip text.

### setAllowXDragLeft

public void setAllowXDragLeft (`Boolean b`)

Sets a flag that determines whether the element's start date can be dragged left or right therefore adjusting the duration of the task. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is `false`.

**Parameters:**

`b` - `true` if the element's start date can be adjusted; `false` otherwise.

## setAllowXDragRight

public void setAllowXDragRight (Boolean b)

Sets a flag that determines whether the element's end date can be dragged left or right therefore adjusting the duration of the task. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is false.

**Parameters:**

b - true if the element's end date can be adjusted; false otherwise.

## setAllowXMove

public void setAllowXMove (Boolean b)

Sets a flag that determines whether the element can be moved to start at a different time. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is false.

**Parameters:**

b - true if the element can be moved horizontally; false otherwise.

## setAllowYMove

public void setAllowYMove (Boolean b)

Sets a flag that determines whether the element can be dragged vertically on the timeline. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is false.

**Parameters:**

b - true if the element can be moved vertically; false otherwise.

## setAllowYMovePredecessor

public void setAllowYMovePredecessor (Boolean b)

Sets a flag that determines whether a dashed relationship line can be drawn from this element interactively on the timeline. The effect of this behavior is controlled by the script include that handles the appropriate event. The default value for this property is false.

**Parameters:**

b - true if a relationship line can be drawn **from** this element; false otherwise.

## setInnerSegmentClass

public void setInnerSegmentClass (String s)

Specifies the name of the class to use for stylizing the inner segment if it exists. The default value is green.

**Parameters:**

s - String that specifies one of the following values:

- green 🟩
- blue 🟦
- silver ⬜

### setInnerSegmentTimeSpan

public void setInnerSegmentTimeSpan (`String startTimeMs, String endTimeMs)`

Creates an inner segment to show within the current timespan defined by the range specified.

**Parameters:**

`startTimeMs` - String that specifies the start time in milliseconds.

`endTimeMs` - String that specifies the end time in milliseconds.

### setInnerSegmentTimeSpan

public void setInnerSegmentTimeSpan (`Number startTimeMs, Number endTimeMs)`

Creates an inner segment to show within the current timespan defined by the range specified.

**Parameters:**

`startTimeMs` - Number that specifies the start time in milliseconds.

`endTimeMs` - Number that specifies the end time in milliseconds.

### setIsChanged

public void setIsChanged (`Boolean b)`

Sets a flag that specifies whether or not the current timeline item has been modified after initialization. The default value for this property is `false`.

**Parameters:**

`b` - `true` if the current element is marked as **changed**; `false` otherwise.

### setPointIconClass

public void setPointIconClass (`String s)`

Sets the icon class to use for displaying the current element on the timeline if the current instance has zero duration. Note that this only affects the current `TimelineSpan` object and will take precedence over the defaultPointIconClass specified by the `GlideTimeline`.

**Parameters:**

`s` - String that specifies one of the following values:

- `milestone` ◆
- `blue_square` 🟦
- `sepia_square` 🟫
- `green_square` 🟩
- `red_square` 🟥
- `black_square` ⬛
- `blue_circle` 🔵
- `sepia_circle` 🟤
- `green_circle` 🟢
- `red_circle` 🔴
- `black_circle` ⚫

### setSpanColor

public void setSpanColor `(String s)`

Sets a flag that specifies whether or not the current timeline item has been modified after initialization. The default value for this property is `false`.

**Parameters:**

`s` - String that specifies the HTML color code.

### setSpanText

public void setSpanText `(String s)`

Sets the text to display adjacent to the time element. Note that this text will only be displayed if the GlideTimeline object has enabled timeline text via `glideTimeline.showTimelineText(true).`

**Parameters:**

`s` - String that specifies the description text.

### setTimeSpan

public void setTimeSpan `(long startTimeMs, long endTimeMs)`

Sets the start and end dates for the current span.

**Parameters:**

`startTimeMs` - A number that specifies the start time in milliseconds.

`endTimeMs` - A number that specifies the end time in milliseconds.

### setTimeSpan

public void setTimeSpan `(String startTimeMs, String endTimeMs)`

Sets the start and end dates for the current span.

**Parameters:**

`startTimeMs` - A string that specifies the start time in milliseconds.

`endTimeMs` - A string that specifies the end time in milliseconds.

### setTooltip

public void setTooltip `(String s)`

Sets the text to display in the tooltip when the TimeSpan element is being hovered over.

**Note:** *You can specify valid HTML in the string that sets the tooltip.*

**Parameters:**

`s` - A string that specifies the tooltip text.

# Example

The following example demonstrates how to create a timeline schedule page with corresponding script include utilizing a majority of the API described above. For this example we are going to create an Incident Summary Timeline for a project support manager to visualize all of the new incidents. All new incidents should be displayed as single points where the priority of the incident is distinguished by a different point icon. Additionally, all closed incidents should be displayed on the timeline in a separate group that shows the duration of the incident before it was closed. Since the Project Manager wants to be able to easily close new items that are resolved without using any form lists, we will handle the vertical move event allowing the new incidents to be dragged into the closed incident group or items within.

## Schedule Page

Create a new Schedule Page with the following properties:

- **Name:** Hardware Incidents
- **Schedule type:** incident_timeline
- **View Type:** Timeline
- **Client Script:**

```
// Set our page configuration
glideTimeline.setReadOnly(false);
glideTimeline.showLeftPane(true);
glideTimeline.showLeftPaneAsTree(true);
glideTimeline.showTimelineText(true);
glideTimeline.showDependencyLines(false);
glideTimeline.groupByParent(true);
glideTimeline.setDefaultPointIconClass('milestone');

// We will define what items to display and provide a custom event
handler for moving new items to the closed state
glideTimeline.registerEvent('getItems',
'IncidentTimelineScriptInclude');
glideTimeline.registerEvent('elementMoveY',
'IncidentTimelineScriptInclude');
```

## Script Include

Now that the schedule page has been created we need to generate a matching script include for the events that were registered. Create a new script include with the following properties:

- **Name:** IncidentTimelineScriptInclude
- **Active:** Checked
- **Client callable:** Checked
- **Script:**

```
// Class Imports

var IncidentTimelineScriptInclude = Class.create();
IncidentTimelineScriptInclude.prototype =
Object.extendsObject(AbstractTimelineSchedulePage, {
```

```
///////////////////////////////////////////////////////////////////////////////////////////////////
  // GET_ITEMS

///////////////////////////////////////////////////////////////////////////////////////////////////
  getItems: function() {
    // Specify the page title
    this.setPageTitle('My Custom Incident Summary Timeline');

    var groupNew = new GlideTimelineItem('new');
    groupNew.setLeftLabelText('New Incidents');
    groupNew.setImage('../images/icons/all.gifx');
    groupNew.setTextBold(true);
    this.add(groupNew);

    var groupClosed = new GlideTimelineItem('closed');
    groupClosed.setLeftLabelText('Closed Incidents');
    groupClosed.setImage('../images/icons/all.gifx');
    groupClosed.setTextBold(true);
    groupClosed.setIsDroppable(true); // This allows us to drag an open
 incident onto the closed group row.
    this.add(groupClosed);

    // Get all the incidents and let's add only the new/closed ones
appropriately
    var gr = new GlideRecord('incident');
    gr.query();
    while (gr.next()) {
      // Only loop through new/closed incidents
      if (gr.incident_state != '1' && gr.incident_state != '7')
        continue;

      // Ok, we have a new/closed incident. Create the item and the
span first.
      var item = new GlideTimelineItem(gr.getTableName(), gr.sys_id);
      var span = item.createTimelineSpan(gr.getTableName(), gr.sys_id);

      // Specific properties for a new incident
      if (gr.incident_state == '1') { // New
        item.setParent(groupNew.getSysId());
        item.setImage('../images/icons/open.gifx');

span.setTimeSpan(gr.getElement('opened_at').getGlideObject().getNumericValue(),

gr.getElement('opened_at').getGlideObject().getNumericValue());
```

```
        // We'll show different colors based upon the priorities only
for new incidents
        switch (gr.getElement('priority').toString()) {
          case '1':
            span.setPointIconClass('red_circle');
            break;
          case '2':
            span.setPointIconClass('red_square');
            break;
          case '3':
            span.setPointIconClass('blue_circle');
            break;
          case '4':
            span.setPointIconClass('blue_square');
            break;
          case '5':
            span.setPointIconClass('sepia_circle');
            break;
          default: // Otherwise, the default point icon class will be
used (Milestone)
        }

      // Specific properties for a closed incident
      } else if (gr.incident_state == '7') {
        item.setParent(groupClosed.getSysId());
        item.setImage('../images/icons/closed.gifx');

span.setTimeSpan(gr.getElement('opened_at').getGlideObject().getNumericValue(),

gr.getElement('closed_at').getGlideObject().getNumericValue());
      }

      // Common item properties
      item.setLeftLabelText(gr.short_description);

      // Common span properties
      span.setSpanText(gr.short_description);
      span.setTooltip('<strong>' +
GlideStringUtil.escapeHTML(gr.short_description) + '</strong><br>' + gr.number);
      span.setAllowXMove(false);
      span.setAllowYMove(gr.canWrite() ? true : false);
      span.setAllowYMovePredecessor(false);
      span.setAllowXDragLeft(false);
      span.setAllowXDragRight(false);

      // Now we add the actual item
      this.add(item);
```

```
    }
  },



//////////////////////////////////////////////////////////////////////////////////////////////////
  // ELEMENT_MOVE_Y

//////////////////////////////////////////////////////////////////////////////////////////////////

  /**
   * This is one of the AbstractTimelineSchedulePage event handler
methods that corresponds to a vertical
   * move. The arguments for this function are defined in the API
section of the event handler methods.
   */
  elementMoveY: function(spanId, itemId, newItemId) {

    // Get information about the current incident
    var gr = new GlideRecord('incident');
    gr.addQuery('sys_id', spanId);
    gr.query();
    if (!gr.next())
      return this.setStatusError('Error', 'Unable to lookup the current
 incident.');

    // Only allow the new incidents to have their state adjusted.
    if (gr.incident_state != '1')
      return this.setStatusError('Error', 'Only new incidents can have
their state adjusted.');

    // Get information about the dropped TimelineItem. If it was
dropped in an item on the "New Incidents"
    // group let's do nothing. If it was dropped in the "Closed
Incidents" then let's adjust the state automatically.
    var grDropped = new GlideRecord('incident');
    grDropped.addQuery('sys_id', newItemId);
    grDropped.query();
    if (!grDropped.next() || grDropped.incident_state == '7') {
      // This means the dropped item was either the 'Closed Incidents'
group (which has no record or sys_id) or an
      // existing incident that is closed. The 'New Incidents' also has
 no sys_id; however, the default behavior for
      // items without a sysId is to be non-droppable. This is why we
explicitly denoted the 'Closed Incidents' to
      // be marked as "droppable".
      // Return a dialog prompt
      this.setStatusPrompt('Confirm',
        'Are you sure you want to close: ' +
          '<div style="margin:10px 0 10px 14px;padding:4px;background-color:#EBEBEB;"><strong>' +
          GlideStringUtil.escapeHTML(gr.short_description) +
          '</strong><br /><div class="font_smaller">' + gr.number + '</div></div>',
        'this._elementMoveYHandler_DoClose',     // This function is
for when the OK button is clicked.
        'this._elementMoveYHandler_DoNothing',   // This function is
for when the Cancel button is clicked.
        'this._elementMoveYHandler_DoNothing');  // This function is
for when the Close button is clicked.
    }
  },

  _elementMoveYHandler_DoClose: function(spanId, itemId, newItemId) {
    // Notice that this function takes the same function arguments as
the original function for which it
    // is a custom event handler for.

    // Update the database record from 'New' to 'Closed'.
    var gr = new GlideRecord('incident');
    gr.addQuery('sys_id', spanId);
    gr.query();
    gr.next();
    gr.setValue('incident_state', '7');
    gr.update();

    // This will re-render the timeline showing the updated item in the
 closed group.
    this.setDoReRenderTimeline(true);

    // Let's show a success message
    this.setStatusSuccess('Success', '<strong>' + gr.short_description + '</strong> was
successfully closed.');
  },

  // Since the user clicked cancel or close we simply do nothing.
  _elementMoveYHandler_DoNothing: function(spanId, itemId, newItemId)
{}

});
```

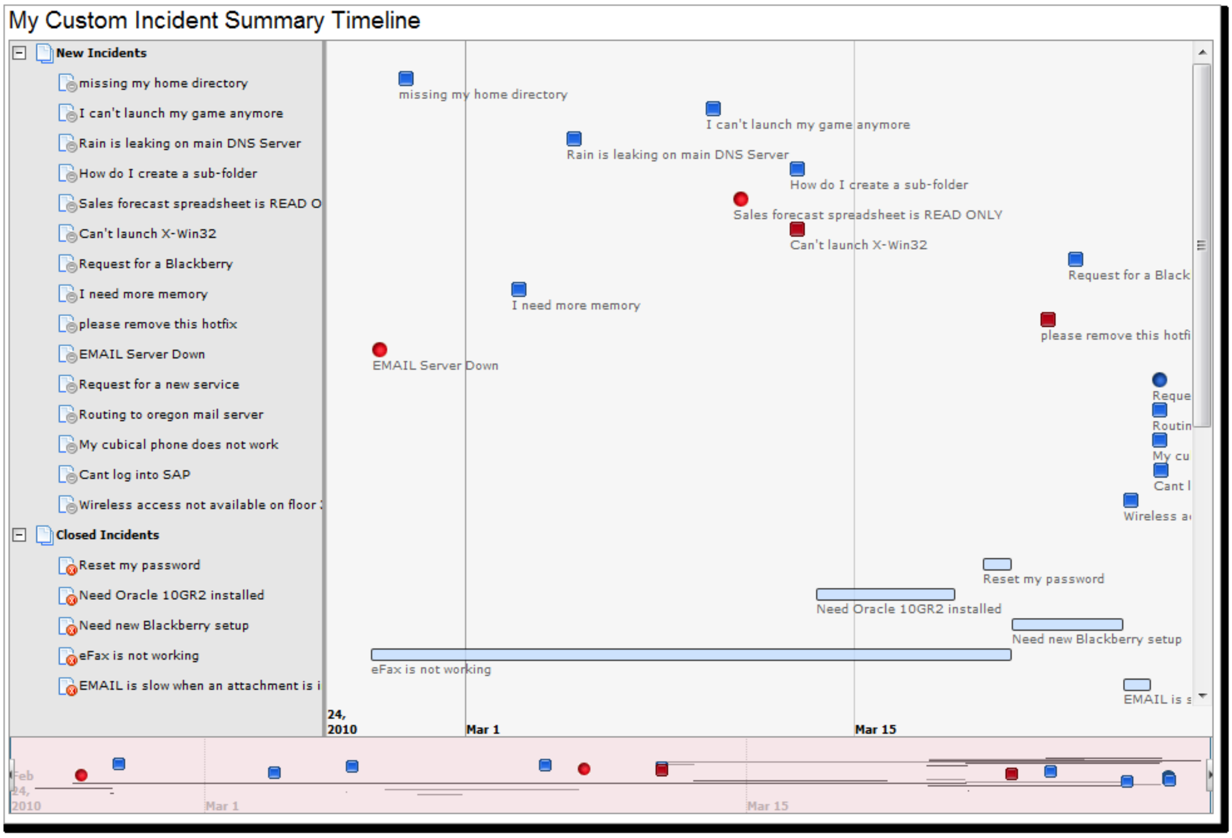
## Screenshots / Results

1. Navigate to:

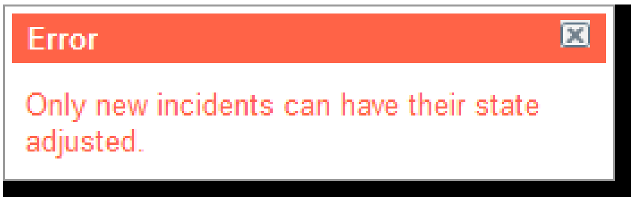
   http://[YOURINSTANCE]:8080/show_schedule_page.do?sysparm_page_schedule_type=**incident_timeline**

   Notice the bold text is the value of the schedule page **Schedule type** field.

2. The page displays a timeline as specified by the schedule page and script include created. A link to this page can be created and placed as a Module or UI Action as necessary.

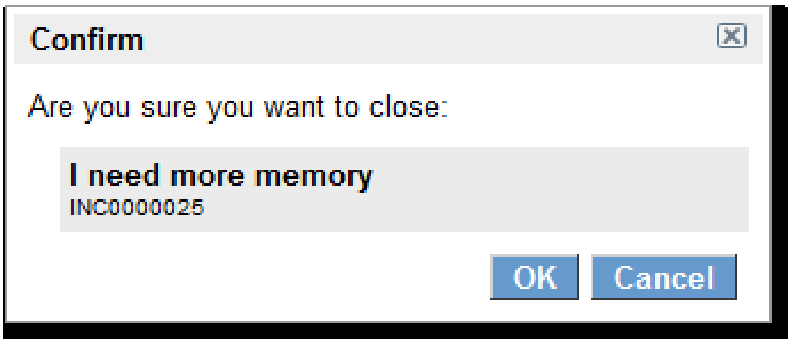

My Custom Incident Summary Timeline
New Incidents
missing my home directory
I can't launch my game anymore
Rain is leaking on main DNS Server
How do I create a sub-folder
Sales forecast spreadsheet is READ O
Can't launch X-Win32
Request for a Blackberry
I need more memory
please remove this hotfix
EMAIL Server Down
Request for a new service
Routing to oregon mail server
My cubical phone does not work
Cant log into SAP
Wireless access not available on floor
Closed Incidents
Reset my password
Need Oracle 10GR2 installed
Need new Blackberry setup
eFax is not working
EMAIL is slow when an attachment is i
missing my home directory
I can't launch my game anymore
Rain is leaking on main DNS Server
How do I create a sub-folder
Sales forecast spreadsheet is READ ONLY
Can't launch X-Win32
Request for a Black
I need more memory
please remove this hotfi
EMAIL Server Down
Reque
Routin
My cu
Cant l
Wireless a
Reset my password
Need Oracle 10GR2 installed
Need new Blackberry setup
eFax is not working
EMAIL is s
24, 2010
Mar 1
Mar 15
Feb 24, 2010
Mar 1
Mar 15

3. Attempting to move a closed incident anywhere displays the expected error message.



Only new incidents can have their state adjusted.

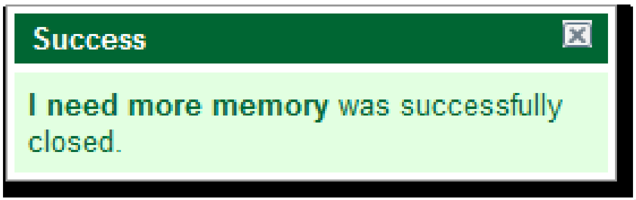
4. Moving the incident: **I need more memory** displays the following confirmation box.



Are you sure you want to close:

I need more memory
INC0000025

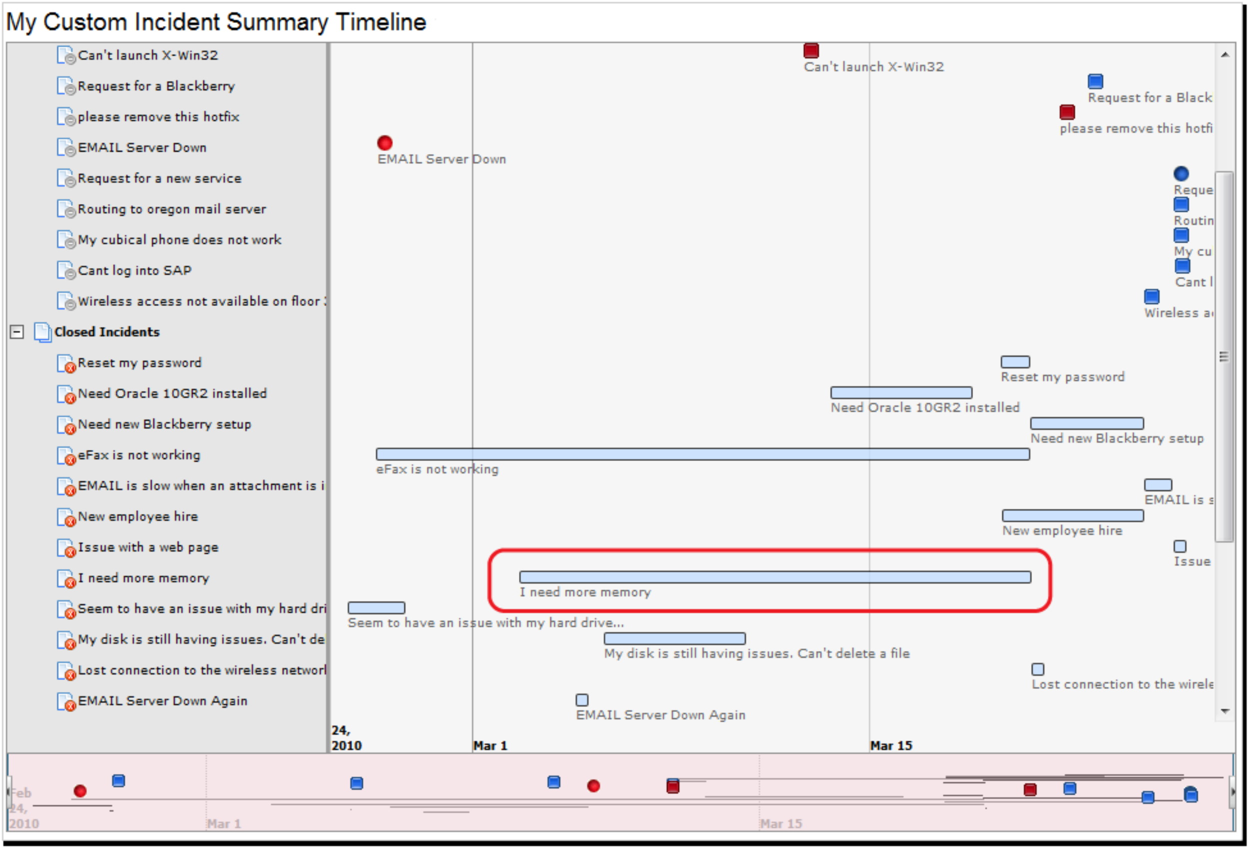
5. Clicking the Cancel button closes the overlay. Clicking the OK button actually updates the incident_state of the record and then displays the following success box.



Success

I need more memory was successfully closed.

6. After clicking OK, it is clear the incident is now listed in the **Closed Incidents** group.



My Custom Incident Summary Timeline
Can't launch X-Win32
Request for a Blackberry
please remove this hotfix
EMAIL Server Down
Request for a new service
Routing to oregon mail server
My cubical phone does not work
Cant log into SAP
Wireless access not available on floor
Closed Incidents
Reset my password
Need Oracle 10GR2 installed
Need new Blackberry setup
eFax is not working
EMAIL is slow when an attachment is i
New employee hire
Issue with a web page
I need more memory
Seem to have an issue with my hard dri
My disk is still having issues. Can't de
Lost connection to the wireless networl
EMAIL Server Down Again
Can't launch X-Win32
Request for a Black
please remove this hotfi
EMAIL Server Down
Reque
Routin
My cu
Cant l
Wireless a
Reset my password
Need Oracle 10GR2 installed
Need new Blackberry setup
eFax is not working
EMAIL is s
New employee hire
Issue
I need more memory
Seem to have an issue with my hard drive...
My disk is still having issues. Can't delete a file
Lost connection to the wirele
EMAIL Server Down Again
24, 2010
Mar 1
Mar 15
Feb 24, 2010
Mar 1
Mar 15

## References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/script/server-api/concept/c_SchedulePages.html

# Time Cards

**Note:** *This article applies to Fuji. For more current information, see Time Cards* [1] *at* http://docs.servicenow.com The ServiceNow Wiki is no longer being updated. Please refer to http://docs.servicenow.com for the latest product documentation.

## Overview

The time card management feature works with the Task table to record time worked on Projects, Incidents, Problems, and Change Requests. Task assignees can record time worked in the **Time worked** field on a task record or enter hours directly into their time card. Some tables support automatic time card creation based on start and end date fields.

Time cards also have an optional approval mechanism for project managers. Administrators and other roles that act as approvers can see all the time cards for the week. All users who are in a role that is responsible for working on tasks also can access their personal time cards. Time cards can have any of the following states.

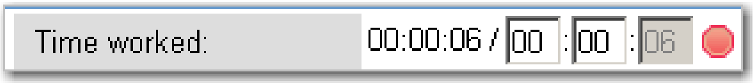
- **Pending**
- **Approved**
- **Submitted**
- **Rejected**

**Note:** *The Time card management plugin is required to use time cards. Some of the procedures on this page require the project management feature, which activates time cards automatically.*

## Recording Time Worked

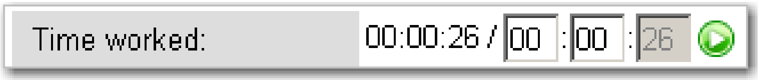
Time accrued on a project or spent working on any record in the Task table is retrieved by the time card from the **Time worked** field. This field is present on Project Task records by default, but does not appear on the Incident, Problem, and Change forms and must be added by configuring the form. Time recorded in this field is used to populate an existing time card or to create a new time card if one does not already exist. This behavior is controlled by a time card **property**. The **Time worked** field has a counter that acts like a stopwatch for the duration of the time spent in the record. The counter can be stopped and started by a button in the field. By default, the **Time worked** counter is enabled and begins recording the elapsed time when the record is opened. Stop the counter with the red stop sign button and restart it with the green *play* button.

Time counter started:



Time worked: 00:00:06 / 00 : 00 : 06

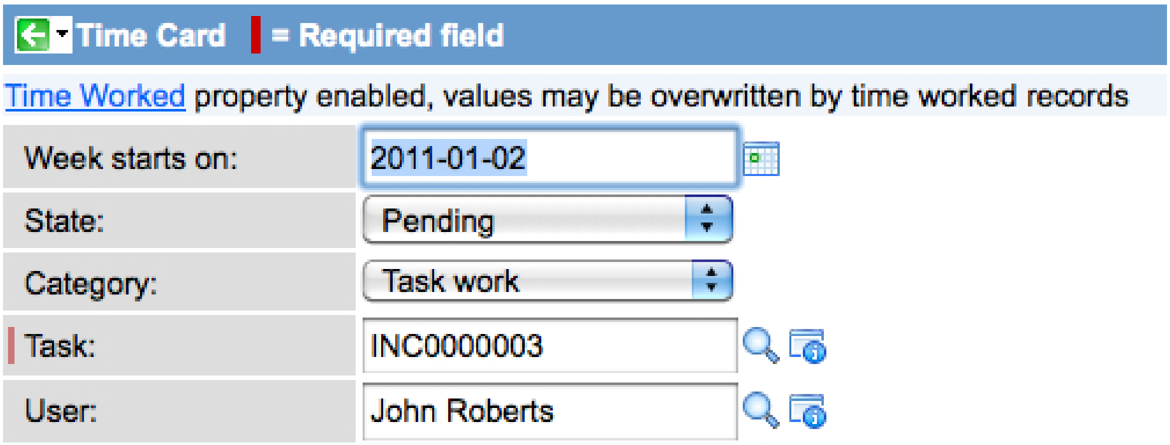
Time counter stopped:



Time worked: 00:00:26 / 00 : 00 : 26

If you are creating time cards from time worked entries, you can add the related list to display the time worked records on the time card form.

You will also notice an informational message on the time card to let you know that changes to time worked records will override values in the time card. This is displayed using a formatter, which can be added or removed by configuring the form.



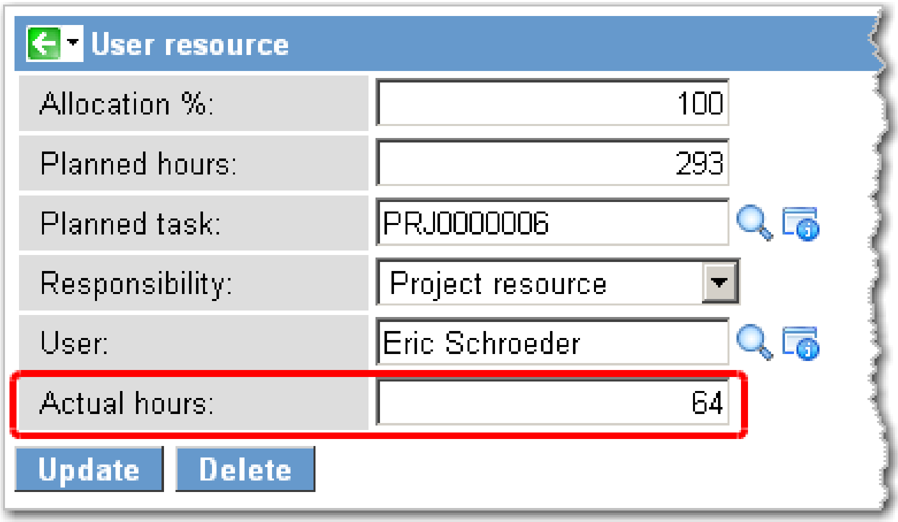
Time Worked property enabled, values may be overwritten by time worked records

| | |
|---|---|
| Week starts on: | 2011-01-02 |
| State: | Pending |
| Category: | Task work |
| Task: | INC0000003 |
| User: | John Roberts |

## Project Management User Resources

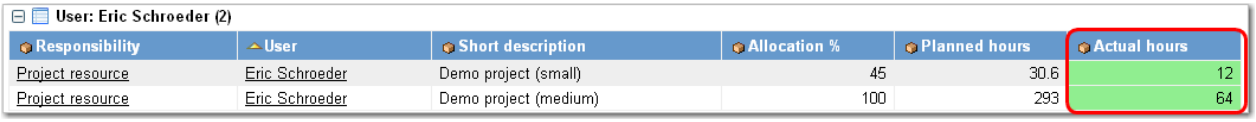
When the project management feature is enabled, the total time from each time card is displayed in the User Resource record for that Project Management task. To view a resource record, navigate to either of the following locations:

- **Project > Views > Resources by Project**:



| | |
|---|---|
| Allocation %: | 100 |
| Planned hours: | 293 |
| Planned task: | PRJ0000006 |
| Responsibility: | Project resource |
| User: | Eric Schroeder |
| Actual hours: | 64 |

- **Project > Views > Project by Resources**:



User: Eric Schroeder (2)

| Responsibility | User | Short description | Allocation % | Planned hours | Actual hours |
|---|---|---|---|---|---|
| Project resource | Eric Schroeder | Demo project (small) | 45 | 30.6 | 12 |
| Project resource | Eric Schroeder | Demo project (medium) | 100 | 293 | 64 |

## Creating a Time Card

Time cards can be created automatically or manually:

- **Automatic**: Configure time cards to be created when a user updates a task record. This behavior is controlled by a time card property that is set to *false* by default. See the table of properties in this page for details. In Incident, Problem, and Change records, the **Time worked** field must be added to the form.
- **Manual**: Create a new time card for each task and enter the times manually.

**Note:** *Time cards cannot be created automatically when you use the mobile interface. Use the desktop interface if you want to use the automatic time card feature.*

Users with the timecard_admin role can create a time card manually:

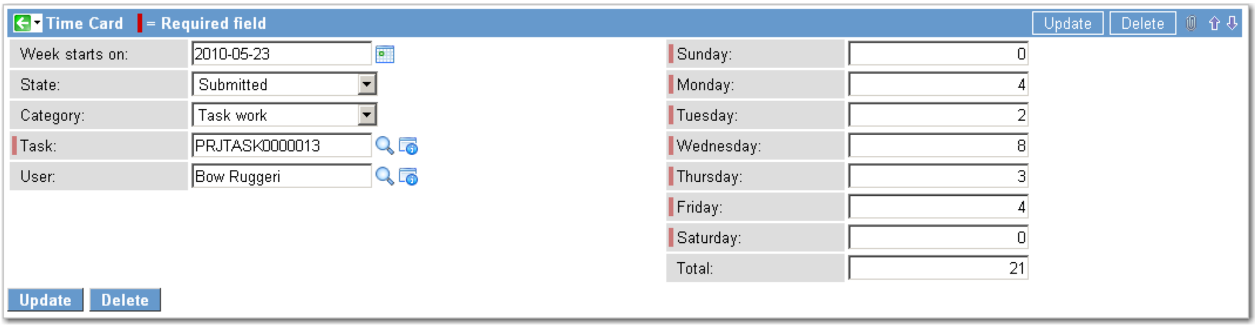
1. Navigate to **Time Cards > All** and click **New**.

   The **Week starts on**, **State**, and **Category** fields are completed automatically. The category defaults to **Task work**, but can be any of the following:

   - Task work
   - Admin
   - Meeting
   - KTLO (maintenance of existing system)
   - Out of office
   - Training
2. Select a **Task** from the pop-up list.

   This can be anything from the Task table.
3. Select your name from the list in the **User** field.
4. Click **Submit**.

   After the time card is created, the hours for that task can be incremented automatically from the **Time worked** field in the task record. This is controlled by a time card property, which is set to *true* by default. See the table of properties in this page for details. If automatic updates are not configured, the time card must be updated manually by the user or an administrator.



| | | | |
|---|---|---|---|
| Week starts on: | 2010-05-23 | Sunday: | 0 |
| State: | Submitted | Monday: | 4 |
| Category: | Task work | Tuesday: | 2 |
| Task: | PRJTASK0000013 | Wednesday: | 8 |
| User: | Bow Ruggeri | Thursday: | 3 |
| | | Friday: | 4 |
| | | Saturday: | 0 |
| | | Total: | 21 |

## Managing Time Cards

The **My Time Cards > Current** module presents a page showing all of your time cards for the current week. There is also a control to **Generate Task Cards**. This button will search for all planned tasks that are scheduled for the current time card period, if you don't already have a time card for the task.

# Properties

Users with the timecard_admin role can set time card properties by navigating to **Time Cards > Administration > Properties**.

| Name | Description | Default |
|---|---|---|
| com.snc.time_card.autocreate | Auto-create a user's time card when they update a task | No |
| com.snc.time_card.time_worked | Auto-fill a user's time card with time from their 'Time worked' entries | No |
| com.snc.time_card.update.effort | Update the task's 'Actual effort' based on the hours entered in the time card | No |
| com.snc.time_card.update.resource | Update the project/user's resource allocation record based on the hours entered in the time card | No |
| com.snc.time_card.start_day | What day should time cards start on, default is Sunday. Changing this value may create duplicate time cards for the week of the change, since time card queries are based on this value. | Sunday |

# Managing Costs

When the cost management feature is enabled, time cards can be used to manage the cost of labor in the **Financial Management** application.

When a time card is marked **Approved**, the user's rate (listed in the **Labor Rate Card**) is used to generate a one-time Expense Line for the time worked. If no Labor Rate Cards apply to the user, the property **com.snc.time_card.default_rate** defines a default rate.

# Roles

The timecard_admin role enables users to approve, modify, and delete the time cards of other users.

# Activating Time Card Management

Administrators can activate the Time card management plugin.

**Click the plus to expand instructions for activating a plugin.**

If you have the admin role, use the following steps to activate the plugin.

1. Navigate to **System Definition > Plugins**.
2. Right-click the plugin name on the list and select **Activate/Upgrade**.

   If the plugin depends on other plugins, these plugins are listed along with their activation status.
3. [Optional] If available, select the **Load demo data** check box.

   Some plugins include demo data—sample records that are designed to illustrate plugin features for common use cases. Loading demo data is a good policy when you first activate the plugin on a development or test instance. You can load demo data after the plugin is activated by repeating this process and selecting the check box.
4. Click **Activate**.

## References

[1] https://docs.servicenow.com/bundle/jakarta-servicenow-platform/page/administer/task-table/concept/c_TimeCards.html

# Article Sources and Contributors

**Project Management** *Source*: http://wiki.servicenow.com/index.php?oldid=250789 *Contributors*: Cheryl.dolan, Davida.hughes, Emily.partridge, Fuji.publishing.user, Guy.yedwab, Jerrod.bennett, John.ramos, John.roberts, Joseph.messerschmidt, Liz.malone, Mark.johnson, Michael.randall, Mike.cribier, Phillip.salzman, Prasad.Rao, Rachel.sienko, Steven.wood, Suzanne.smith, Wallymarx

**Installed with Project Management** *Source*: http://wiki.servicenow.com/index.php?oldid=247299 *Contributors*: Cheryl.dolan, David.Bailey, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.roberts, Joseph.messerschmidt, Liz.malone, Mark.johnson, Michael.randall, Phillip.salzman, Steven.wood

**Project Management v2 to v3 Upgrade** *Source*: http://wiki.servicenow.com/index.php?oldid=239567 *Contributors*: Joseph.messerschmidt, Phillip.salzman

**Project Management v1 to v2 Upgrade** *Source*: http://wiki.servicenow.com/index.php?oldid=241415 *Contributors*: CapaJC, Carleen.greenlee, Emily.partridge, Fuji.publishing.user, G.yedwab, Jerrod.bennett, Joey.mart, Joseph.messerschmidt, Phillip.salzman, Rachel.sienko, Steven.wood

**Proposing a New Project** *Source*: http://wiki.servicenow.com/index.php?oldid=189661 *Contributors*: Phillip.salzman

**Creating a Project** *Source*: http://wiki.servicenow.com/index.php?oldid=250160 *Contributors*: Cheryl.dolan, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, Joseph.messerschmidt, Liz.malone, Michael.randall, Phillip.salzman, Steven.wood, Suzanne.smith

**Creating Project Tasks** *Source*: http://wiki.servicenow.com/index.php?oldid=250448 *Contributors*: Cheryl.dolan, David.Bailey, Fuji.publishing.user, G.yedwab, John.ramos, Joseph.messerschmidt, Liz.malone, Mark.stanger, Michael.randall, Phillip.salzman, Prakash.chandra, Rachel.sienko, Steven.wood, Suzanne.smith

**Project Task Relationships and Dependencies** *Source*: http://wiki.servicenow.com/index.php?oldid=250795 *Contributors*: Cheryl.dolan, Fuji.publishing.user, John.ramos, Phillip.salzman, Prakash.chandra, Rachel.sienko

**Gantt Chart** *Source*: http://wiki.servicenow.com/index.php?oldid=250588 *Contributors*: Cheryl.dolan, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, Joseph.messerschmidt, Phillip.salzman, Steven.wood

**Work Breakdown Structure** *Source*: http://wiki.servicenow.com/index.php?oldid=251057 *Contributors*: Cheryl.dolan, Emily.partridge, Fuji.publishing.user, John.ramos, Phillip.salzman, Publishing.user

**Task Resources** *Source*: http://wiki.servicenow.com/index.php?oldid=250944 *Contributors*: G.yedwab, Guy.yedwab, John.ramos, John.roberts, Joseph.messerschmidt, Phillip.salzman, Publishing.user, Steven.wood

**Starting a Project** *Source*: http://wiki.servicenow.com/index.php?oldid=189669 *Contributors*: Emily.partridge, Phillip.salzman

**Project Portfolio Management** *Source*: http://wiki.servicenow.com/index.php?oldid=250791 *Contributors*: CapaJC, Cheryl.dolan, Emily.partridge, Fuji.publishing.user, Guy.yedwab, John.ramos, Joseph.messerschmidt, Liz.malone, Michael.randall, Phillip.salzman, Prakash.chandra, Rachel.sienko, Steven.wood, Suzanne.smith, Vaughn.romero, Wallymarx

**Project Portfolio Management Security** *Source*: http://wiki.servicenow.com/index.php?oldid=238795 *Contributors*: Cheryl.dolan, Davida.hughes, Emily.partridge, Fuji.publishing.user, Guy.yedwab, Jerrod.bennett, John.ramos, John.roberts, Joseph.messerschmidt, Liz.malone, Mark.johnson, Michael.randall, Mike.cribier, Phillip.salzman, Prasad.Rao, Rachel.sienko, Steven.wood, Suzanne.smith, Wallymarx

**Project Reporting** *Source*: http://wiki.servicenow.com/index.php?oldid=250793 *Contributors*: Fuji.publishing.user, John.ramos, Liz.malone, Phillip.salzman, Publishing.user

**Updating a Project in Progress** *Source*: http://wiki.servicenow.com/index.php?oldid=241922 *Contributors*: Fuji.publishing.user, Liz.malone, Phillip.salzman, Suzanne.smith

**Closing a Project** *Source*: http://wiki.servicenow.com/index.php?oldid=194116 *Contributors*: Cheryl.dolan, David.Bailey, Phillip.salzman

**Project Management v2 Costing Add-on** *Source*: http://wiki.servicenow.com/index.php?oldid=89896 *Contributors*: Emily.partridge, G.yedwab, Guy.yedwab, Joe.Westrich, John.roberts, Joseph.messerschmidt, Phillip.salzman, Rachel.sienko, Steven.wood

**Managing Project Costs** *Source*: http://wiki.servicenow.com/index.php?oldid=250711 *Contributors*: G.yedwab, Guy.yedwab, Joe.Westrich, John.ramos, John.roberts, Joseph.messerschmidt, Phillip.salzman, Steven.wood

**Project Management FAQ** *Source*: http://wiki.servicenow.com/index.php?oldid=250790 *Contributors*: Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, John.ramos, Joseph.messerschmidt, Liz.malone, Phillip.salzman, Publishing.user, Steven.wood

**Project Management Processing Rules** *Source*: http://wiki.servicenow.com/index.php?oldid=150368 *Contributors*: Cheryl.dolan, Fuji.publishing.user, John.ramos, Phillip.salzman, Prakash.chandra, Rachel.sienko

**Schedule Pages** *Source*: http://wiki.servicenow.com/index.php?oldid=250837 *Contributors*: Emily.partridge, Eric.jacobson, G.yedwab, Gadi.yedwab, George.rawlins, Guy.yedwab, Jim.uebbing, Joe.Westrich, John.ramos, Joseph.messerschmidt, Mark.johnson, Neola, Rachel.sienko, Steven.wood

**Time Cards** *Source*: http://wiki.servicenow.com/index.php?oldid=250381 *Contributors*: Davida.hughes, Emily.partridge, Fuji.publishing.user, G.yedwab, Guy.yedwab, Jerrod.bennett, Joe.Westrich, John.ramos, John.roberts, Joseph.messerschmidt, Phillip.salzman, Rachel.sienko, Steven.wood

# Image Sources, Licenses and Contributors

**Image:Warning.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Warning.gif *License*: unknown *Contributors*: CapaJC

**Image:project_phases.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Project_phases.png *License*: unknown *Contributors*: Phillip.salzman

**Image:ProjectMenuFuji.png** *Source*: http://wiki.servicenow.com/index.php?title=File:ProjectMenuFuji.png *License*: unknown *Contributors*: Fuji.publishing.user

**Image:ProjectMgmtMenuEureka.png** *Source*: http://wiki.servicenow.com/index.php?title=File:ProjectMgmtMenuEureka.png *License*: unknown *Contributors*: Fuji.publishing.user

**Image:migration_v2_v3_PPMlog.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Migration_v2_v3_PPMlog.png *License*: unknown *Contributors*: Phillip.salzman

**Image:migration_v2_v3_Projectlog.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Migration_v2_v3_Projectlog.png *License*: unknown *Contributors*: Phillip.salzman

**File:projectformdefault.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Projectformdefault.png *License*: unknown *Contributors*: Jerrod.bennett

**File:projecttaskformdefault.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Projecttaskformdefault.png *License*: unknown *Contributors*: Jerrod.bennett

**File:projectlistdefault.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Projectlistdefault.png *License*: unknown *Contributors*: Jerrod.bennett

**File:projecttasklistdefault.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Projecttasklistdefault.png *License*: unknown *Contributors*: Jerrod.bennett

**Image:propose_new_project.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Propose_new_project.png *License*: unknown *Contributors*: Phillip.salzman

**Image:propose_new_project_form.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Propose_new_project_form.png *License*: unknown *Contributors*: Phillip.salzman

**Image:CopyProject2.png** *Source*: http://wiki.servicenow.com/index.php?title=File:CopyProject2.png *License*: unknown *Contributors*: Phillip.salzman

**Image:create_project_ui_actions.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Create_project_ui_actions.png *License*: unknown *Contributors*: Phillip.salzman

**Image:create_project_incident.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Create_project_incident.png *License*: unknown *Contributors*: Phillip.salzman

**Image:new_task.png** *Source*: http://wiki.servicenow.com/index.php?title=File:New_task.png *License*: unknown *Contributors*: Phillip.salzman

**Image:related_list_tasks.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Related_list_tasks.png *License*: unknown *Contributors*: Phillip.salzman

**Image:percentage_complete_project.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Percentage_complete_project.png *License*: unknown *Contributors*: Phillip.salzman

**Image:project_task_creator.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Project_task_creator.png *License*: unknown *Contributors*: Phillip.salzman

**Image:copy_partial_project_dublin.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Copy_partial_project_dublin.png *License*: unknown *Contributors*: Phillip.salzman

**Image:insert_project_task_list.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Insert_project_task_list.png *License*: unknown *Contributors*: Phillip.salzman

**Image:milestone_between_phases.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Milestone_between_phases.png *License*: unknown *Contributors*: Phillip.salzman

**Image:skills.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Skills.png *License*: unknown *Contributors*: Phillip.salzman

**Image:notificationwf.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Notificationwf.png *License*: unknown *Contributors*: Jerrod.bennett

**Image:Finish_to_start_dependency.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Finish_to_start_dependency.png *License*: unknown *Contributors*: Phillip.salzman

**image:sucessor_pending_WIP.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Sucessor_pending_WIP.png *License*: unknown *Contributors*: Phillip.salzman

**Image:delete_task_relationship.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Delete_task_relationship.png *License*: unknown *Contributors*: Phillip.salzman

**Image:Parent.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Parent.png *License*: unknown *Contributors*: Phillip.salzman

**Image:remove_parent-child_relationship.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Remove_parent-child_relationship.png *License*: unknown *Contributors*: Phillip.salzman

**image:rollup_popup.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Rollup_popup.png *License*: unknown *Contributors*: Phillip.salzman

**image:Gantt_chart_example.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Gantt_chart_example.png *License*: unknown *Contributors*: Phillip.salzman

**Image:Gannt_chart_tooltip_summary.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Gannt_chart_tooltip_summary.png *License*: unknown *Contributors*: Phillip.salzman

**Image:gantt_chart_drag.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Gantt_chart_drag.png *License*: unknown *Contributors*: Phillip.salzman

**Image:drag_new_task.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Drag_new_task.png *License*: unknown *Contributors*: Phillip.salzman

**Image:gantt_chart_critical_path.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Gantt_chart_critical_path.png *License*: unknown *Contributors*: Phillip.salzman

**Image:WBS_arrow_icon.png** *Source*: http://wiki.servicenow.com/index.php?title=File:WBS_arrow_icon.png *License*: unknown *Contributors*: Phillip.salzman

**Image:WBS.png** *Source*: http://wiki.servicenow.com/index.php?title=File:WBS.png *License*: unknown *Contributors*: Phillip.salzman

**Image:Gear.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Gear.png *License*: unknown *Contributors*: CapaJC

**Image:WBS_gantt.png** *Source*: http://wiki.servicenow.com/index.php?title=File:WBS_gantt.png *License*: unknown *Contributors*: Phillip.salzman

**Image:User_Resource_New.png** *Source*: http://wiki.servicenow.com/index.php?title=File:User_Resource_New.png *License*: unknown *Contributors*: Steven.wood

**Image:User_Resource_Config.png** *Source*: http://wiki.servicenow.com/index.php?title=File:User_Resource_Config.png *License*: unknown *Contributors*: Steven.wood

**Image:Group_Resources_List2.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Group_Resources_List2.png *License*: unknown *Contributors*: Steven.wood

**Image:Group_Resources_List3.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Group_Resources_List3.png *License*: unknown *Contributors*: Steven.wood

**Image:Timeline_Start.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_Start.png *License*: unknown *Contributors*: Steven.wood

**Image:Timeline_Assign_to_User.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_Assign_to_User.png *License*: unknown *Contributors*: Steven.wood

**Image:Timeline_Assigned.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_Assigned.png *License*: unknown *Contributors*: Steven.wood

**Image:Timeline_Detail.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_Detail.png *License*: unknown *Contributors*: Steven.wood

**image:project_start.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Project_start.png *License*: unknown *Contributors*: Phillip.salzman

**Image:PortfolioForm.png** *Source*: http://wiki.servicenow.com/index.php?title=File:PortfolioForm.png *License*: unknown *Contributors*: Fuji.publishing.user

**File:portfolio-demand relationship.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Portfolio-demand_relationship.png *License*: unknown *Contributors*: Prakash.chandra

**File:portfolio-project relationship.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Portfolio-project_relationship.png *License*: unknown *Contributors*: Prakash.chandra

**File:project-portfoliostatus-portfolioprojects.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Project-portfoliostatus-portfolioprojects.png *License*: unknown *Contributors*: Prakash.chandra

**Image:Portfolio_Gantt_Chart2.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Portfolio_Gantt_Chart2.png *License*: unknown *Contributors*: Phillip.salzman

**image:portfolio_dashboard.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Portfolio_dashboard.png *License*: unknown *Contributors*: Phillip.salzman

**Image:FormLiveFeedUI15.png** *Source*: http://wiki.servicenow.com/index.php?title=File:FormLiveFeedUI15.png *License*: unknown *Contributors*: Fuji.publishing.user

**image:baseline_v2.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Baseline_v2.png *License*: unknown *Contributors*: Phillip.salzman

**image:project_triangle.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Project_triangle.png *License*: unknown *Contributors*: Phillip.salzman

**Image:PMC-concepts.png** *Source*: http://wiki.servicenow.com/index.php?title=File:PMC-concepts.png *License*: unknown *Contributors*: G.yedwab

**Image:Group_Hourly_Rate.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Group_Hourly_Rate.png *License*: unknown *Contributors*: Steven.wood

**Image:Group_Resources_List.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Group_Resources_List.png *License*: unknown *Contributors*: Steven.wood

**Image:Group_Resource_Form.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Group_Resource_Form.png *License*: unknown *Contributors*: Steven.wood

**Image:Group_Resources_Estimated_Cost.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Group_Resources_Estimated_Cost.png *License*: unknown *Contributors*: Steven.wood

**Image:PRJ-expenseline.png** *Source*: http://wiki.servicenow.com/index.php?title=File:PRJ-expenseline.png *License*: unknown *Contributors*: G.yedwab

**Image:PRJ-actualcost.png** *Source*: http://wiki.servicenow.com/index.php?title=File:PRJ-actualcost.png *License*: unknown *Contributors*: G.yedwab

**File:TimelineFlow.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineFlow.png *License*: unknown *Contributors*: Mark.johnson

**File:timeline_milestone.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_milestone.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_blue_square.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_blue_square.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_sepia_square.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_sepia_square.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_green_square.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_green_square.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_red_square.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_red_square.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_black_square.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_black_square.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_blue_circle.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_blue_circle.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_sepia_circle.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_sepia_circle.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_green_circle.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_green_circle.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_red_circle.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_red_circle.gif *License*: unknown *Contributors*: Mark.johnson

**File:timeline_black_circle.gif** *Source*: http://wiki.servicenow.com/index.php?title=File:Timeline_black_circle.gif *License*: unknown *Contributors*: Mark.johnson

**File:TimelineInnerSegment_Green.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineInnerSegment_Green.png *License*: unknown *Contributors*: Mark.johnson

**File:TimelineInnerSegment_Blue.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineInnerSegment_Blue.png *License*: unknown *Contributors*: Mark.johnson

**File:TimelineInnerSegment_Silver.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineInnerSegment_Silver.png *License*: unknown *Contributors*: Mark.johnson

**File:TimelineExample-IncidentPreview.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineExample-IncidentPreview.png *License*: unknown *Contributors*: Mark.johnson

**File:TimelineExample-ErrorMoving.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineExample-ErrorMoving.png *License*: unknown *Contributors*: Mark.johnson

**File:TimelineExample-ConfirmClose.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineExample-ConfirmClose.png *License*: unknown *Contributors*: Mark.johnson

**File:TimelineExample-CloseSuccess.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineExample-CloseSuccess.png *License*: unknown *Contributors*: Mark.johnson

**File:TimelineExample-IncidentUpdated.png** *Source*: http://wiki.servicenow.com/index.php?title=File:TimelineExample-IncidentUpdated.png *License*: unknown *Contributors*: Mark.johnson

**Image:Time_Worked_Started.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Time_Worked_Started.png *License*: unknown *Contributors*: Steven.wood

**Image:Time_Worked_Stopped.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Time_Worked_Stopped.png *License*: unknown *Contributors*: Steven.wood

**File:Time_worked_notice.jpg** *Source*: http://wiki.servicenow.com/index.php?title=File:Time_worked_notice.jpg *License*: unknown *Contributors*: John.roberts

**Image:User_Resource.png** *Source*: http://wiki.servicenow.com/index.php?title=File:User_Resource.png *License*: unknown *Contributors*: Steven.wood

**Image:User_Resource2.png** *Source*: http://wiki.servicenow.com/index.php?title=File:User_Resource2.png *License*: unknown *Contributors*: Steven.wood

**Image:Time_Card.png** *Source*: http://wiki.servicenow.com/index.php?title=File:Time_Card.png *License*: unknown *Contributors*: Steven.wood