A light gray grid pattern serves as the background for the title. The grid is composed of small squares that cover the right half of the page.

Integrity

**MKS Integrity 2009 SP6**

---

*User Guide*

---

## MKS Integrity 2009 User Guide

Copyright © 2001–2011 MKS Software Inc.; in Canada copyright owned by MKS Inc. All rights reserved.

MKS makes no warranty of any kind with regard to this material, including, but not limited to the implied warranties of merchantability, performance, or fitness for a particular purpose. MKS shall not be liable for errors contained herein, or for any direct, indirect, incidental, or consequential damages resulting from the use of this material.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means, without written permission from MKS.

MKS, MKS Integrity, Implementer, MKS Toolkit, Sandbox, NuTCRACKER, and MKS Federated Server are trademarks of MKS Inc. All other trademarks are the property of their respective holders.

### Corporate Headquarters

410 Albert Street  
Waterloo, ON N2L 3V3  
Canada  
tel: 519 884 2251  
fax: 519 884 8861  
sales (toll free): 800 265 2797

[www.mks.com](http://www.mks.com)

### Worldwide Offices:

1815 South Meyers Rd.  
Suite 220  
Oakbrook Terrace, IL USA  
60181  
tel: 630 827 4900  
fax: 630 629 9167  
sales (toll free): 800 633 1235  
12701 Fair Lakes Circle  
Suite 350  
Fairfax, VA USA  
22033  
tel: 1 703 803 3343  
fax: 1 703 803 3344  
sales (toll free): 1 800 637 8034  
Martinstraße 42-44  
73728 Esslingen  
Germany  
tel: +49 711 351775 0  
fax: +49 711 351775 7555

Third Floor, Duke's Court  
Duke Street, Woking  
Surrey  
GU21 5BH  
United Kingdom  
tel: +44 (0)1483 733900  
fax: +44 (0)1483 733901  
sales: +44 (0)1483 733919  
3 Killiney Road  
#07-05 Winsland House 1  
Singapore 239519  
tel: +65-6732-8768  
fax: +65-6732-0768  
5th Floor, Unosawa Tokyu Building  
1-19-15, Ebisu, Shibuya-ku,  
Tokyo 150-0013, Japan  
tel: +81-3-5422-9503  
fax: +81-3-5422-9509

This document is uncontrolled when printed or copied.

I2009SP6

**MKS**

---

# Table of Contents

## PART 1: TRACKING CHANGES WITH MKS INTEGRITY ITEMS

What Are Items? .....	2
What Are Item Relationships? .....	3
Items View: Managing Items .....	5
Viewing Item Details .....	9
Searching for Items .....	13
Using Project Filters in the Items View (GUI) .....	17
Filtering Items (Web only) .....	18
Creating an Item .....	21
Editing Items Inline (GUI Only) .....	33
Editing an Item .....	34
Batch Editing Items .....	36
Resolving Conflicts When Editing an Item .....	37
Working With Item Change Packages .....	39
Working With Item Labels .....	40
Copying an Item .....	41
Viewing a Historical Item .....	42
Relationships View: Viewing Item Relationships .....	44
Printing an Item .....	49
Receiving E-mail Notifications About Items .....	50
Recording Effort With Time Entries .....	54
Time Entries View: Managing Time Entries .....	56
Creating a Time Entry .....	57
Editing Time Entries .....	59
Viewing Time Entries .....	60

## PART 2: DISPLAYING ITEMS MEETING SPECIFIC CONDITIONS

What Is a Query? .....	62
Queries View: Managing Your Queries .....	63
Query/Query Definition View .....	64
Query Builder .....	65
Query Filters .....	66
Creating a Query .....	68
Editing a Query .....	77

---

Running a Query .....	78
Copying a Query .....	79
Deleting a Query .....	80
Exporting Query Results to Microsoft Excel .....	81

### **PART 3: DISPLAYING ITEM DATA GRAPHICALLY**

What Is a Chart? .....	86
Charts View: Managing Charts .....	87
Chart Definition View .....	88
Creating a Chart .....	89
Editing a Chart.....	109
Running a Chart .....	110
Copying a Chart.....	112
Saving and Printing a Chart.....	113
Deleting a Chart.....	114

### **PART 4: VIEWING ITEM DATA IN REPORTS**

What Is a Report? .....	116
Reports View: Managing Reports .....	117
Report Definition View.....	118
Creating a Report .....	119
Editing a Report.....	129
Copying a Report.....	130
Running a Report .....	131
Saving and Printing a Report.....	132
Deleting a Report.....	133
Reporting on Items Using Item Reports.....	134

### **PART 5: VIEWING PROJECT DATA IN DASHBOARDS**

What is a Dashboard? .....	137
Designing a Dashboard.....	138
Dashboard Definition View .....	140
Dashboard Layout Designer: User Interface Components .....	141
Dashboards View: Managing Dashboards .....	142
Creating a Dashboard .....	143
Editing a Dashboard .....	150
Running a Dashboard .....	151
Copying a Dashboard .....	153

---

Deleting a Dashboard .....	154
Creating a Dashboard in XML .....	155
<b>PART 6: PERFORMING CALCULATIONS BETWEEN FIELDS</b>	
What are Computed Expressions? .....	159
Computed Expression Rules .....	161
Computed Expression Function Classes .....	165
Creating a Computed Expression in a Report .....	177
Creating a Computed Expression in a Chart .....	179
<b>PART 7: WORKING WITH DOCUMENTS</b>	
What is a Document? .....	181
Document View .....	184
Document Finder .....	189
Working in the Document View .....	190
Creating a Document .....	192
Adding Content to a Document .....	194
Modifying Content .....	199
Finding and Viewing Documents .....	203
Filtering in the Document View .....	204
Printing Documents .....	206
Viewing and Editing Document or Content Properties .....	207
Viewing Document Model Relationships .....	208
Changing the Document Project .....	209
Baselining Documents .....	210
Branching Documents .....	212
Managing Trace Relationships .....	215
Running Current and Historical Reports on Documents .....	217
Propagating Traces to Branched Documents .....	218
Importing and Exporting Documents .....	224
<b>PART 8: EXECUTING TESTS WITH MKS INTEGRITY</b>	
What is Test Management? .....	226
Creating a Test Session from Test Items .....	227
Viewing Test Result Details .....	228
Entering Test Results .....	229
Working With Item Test Results .....	231
Test Results View: Analyzing Test Results .....	233

---

Working With the Test Result Editor .....	234
Editing Test Results.....	237
Purging Test Results .....	239

## PART 9: GROUPING FILES UNDER VERSION CONTROL

Configuration Management Projects View.....	241
Configuration Management Project Information View .....	242
Creating a Configuration Management Project .....	244
Importing a Configuration Management Project .....	245
Dropping a Configuration Management Project.....	247
Adding a Configuration Management Project .....	248
Opening a Configuration Management Project .....	249
Creating a Configuration Management Subproject .....	252
Adding a Configuration Management Subproject .....	253
Adding a Shared Configuration Management Subproject.....	254
Configuring a Configuration Management Subproject.....	256
Dropping a Configuration Management Subproject .....	258
Moving a Configuration Management Subproject .....	259
Viewing a Configuration Management Project History .....	261
Restoring a Configuration Management Project .....	263
Adding Configuration Management Project Labels .....	265
Deleting Configuration Management Project Labels .....	266
Checkpointing a Configuration Management Project .....	267
Comparing Configuration Management Project Checkpoints .....	269
Viewing Configuration Management Project Metrics.....	271
Branching Configuration Management Projects Overview .....	273
Creating a Development Path .....	274
Removing a Development Path.....	275
Merging a Child Development Path.....	276
Locating Where a Configuration Management Object Is Used .....	280
Promoting and Demoting Configuration Management Projects.....	283

## PART 10: MANAGING YOUR PERSONAL WORKSPACE IN A SANDBOX

Sandboxes Overview.....	286
My Sandboxes View: Managing Sandboxes .....	288
Viewing Sandbox Details.....	289
Creating a Sandbox.....	291

---

Configuring a Sandbox .....	294
Opening a Sandbox.....	295
Importing a Sandbox .....	297
Sharing Sandboxes .....	298
Dropping a Sandbox .....	300
Retargeting a Sandbox .....	301
Taking Sandbox Snapshots.....	302

## PART 11: MANAGING SOURCE FILES AS MEMBERS

Adding Members to a Project .....	305
Adding Dropped Members to a Project .....	308
Character Encoding.....	310
Dropping Members From a Project .....	311
Filtering Members .....	313
Importing Members into a Project .....	316
Viewing Member / Revision Information .....	318
Selecting Members .....	320
Moving Members Between Projects .....	322
Identifying Members to Add to Projects.....	326
Viewing and Editing Member Content.....	328
Checking Out a Member .....	329
Checking In a Member .....	334
Renaming a Member.....	339
Discarding Changes to a Member.....	341
Discarding Changes to a Member.....	342
Resyncing Members .....	343
Locking a Member.....	344
Viewing Member Locks .....	345
Removing or Downgrading Locks on Members.....	347
Making a Member's Working File Writable .....	348
Using Keywords in Working Files .....	349
Adding Labels to Members .....	353
Deleting Member Labels .....	354
Freezing Members.....	355
Thawing Members .....	356
Submitting Deferred Operations .....	357
Resyncing Members With Deferred Operations .....	358

---

Cancelling Deferred Operations . . . . .	359
Viewing a Member History . . . . .	360
Filtering a Member or Project History in the GUI . . . . .	364
Viewing Member Archive Information . . . . .	366
Setting the Default Member Revision . . . . .	367
Setting a Revision Rule for Members . . . . .	370
Deleting a Revision . . . . .	373
Viewing Member Changes By Revision . . . . .	374
Working With Pending Revisions . . . . .	376
Comparing Revisions . . . . .	377
Branching Members . . . . .	380
Merging Members . . . . .	381
Merging Modified Working Files . . . . .	383
Resolving Merges . . . . .	384
Working With MKS Visual Difference . . . . .	385
Working With MKS Visual Merge . . . . .	388
Deferring Member Operations . . . . .	391
Generating Member or Project Reports . . . . .	392
Promoting and Demoting Members . . . . .	393

## **PART 12: GROUPING UNITS OF WORK IN CHANGE PACKAGES**

Change Package Overview . . . . .	396
My Change Packages View: Managing Change Packages . . . . .	399
Creating a Change Package . . . . .	400
Discarding ChangePackage Entries . . . . .	402
Viewing ChangePackage Entry Member Differences . . . . .	403
Finding Change Packages . . . . .	404
Viewing Change Package Details and Entries . . . . .	406
Editing a Change Package . . . . .	409
Moving ChangePackage Entries . . . . .	410
Viewing a Generic Change Package . . . . .	412
Submitting Change Packages . . . . .	414
Closing a Change Package . . . . .	415
Discarding Change Packages . . . . .	416
Change Package Reviews Overview . . . . .	417
Reviewing Change Packages . . . . .	422
Reopening a ChangePackage . . . . .	425

---

Apply CP Overview .....	426
Apply CP Procedure .....	433
Resync CP Overview.....	438
Resync CP Procedure.....	446
Apply CP and Resync CP Overview .....	451
Advanced Apply CP and Resync CP Concepts.....	452
Resync By CP Overview .....	456
Integrating With Implementer .....	458
Index .....	459

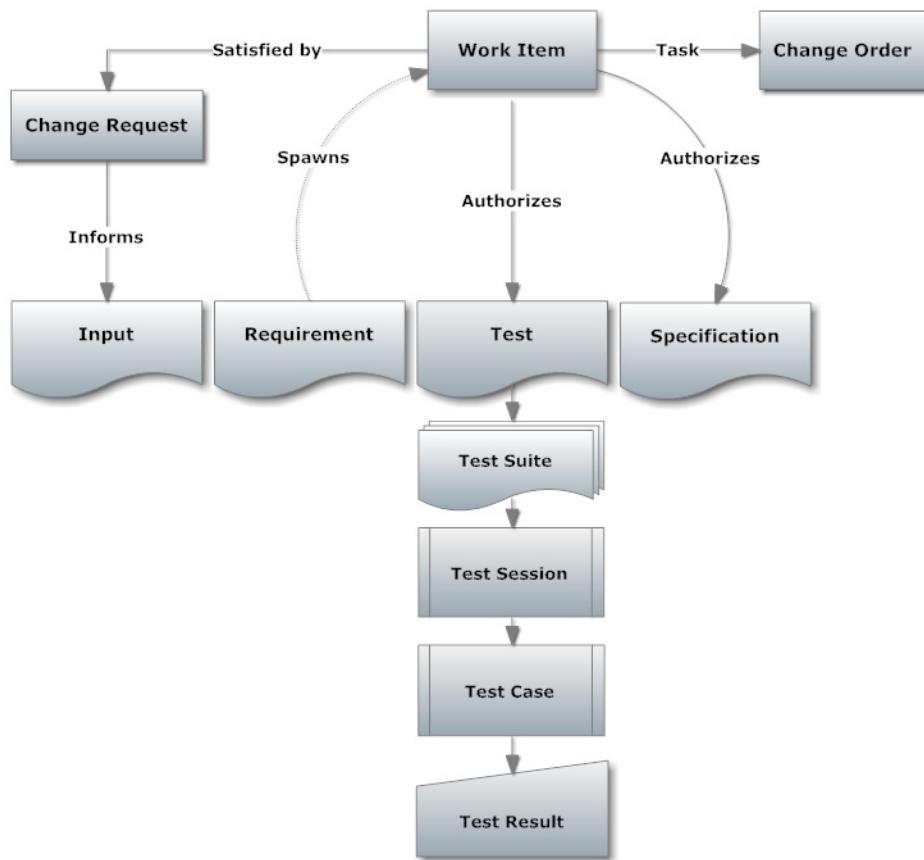


## PART 1

# Tracking Changes With MKS Integrity Items

# What Are Items?

*Items* are the basic building blocks of the MKS Integrity data model, and represent the information assets and process objects associated with software development. For example, items can represent portfolios, projects, requirements, functional specifications, test cases, change requests, and defects. Standard items in MKS Integrity include all of the assets needed to manage the end-to-end development lifecycle. Every item is configurable and contains metadata and history about that item. New item types can be added as needed to address your organization's specific needs.



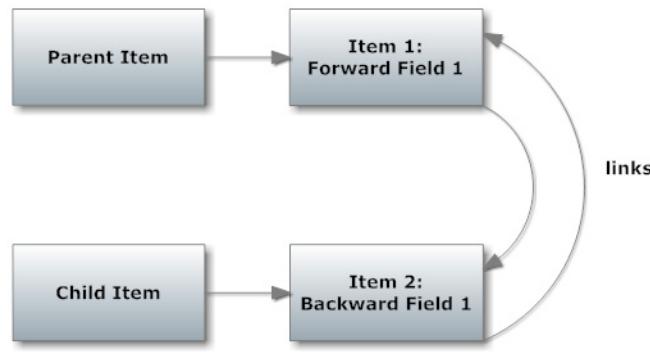
*Item types and how they relate to one another in the Application Lifecycle Management solution*

Every item type has its own enforceable workflow which describes how it moves from state to state, complete with definable rules, full audit trails, and security. Escalations, conditional assignments, notifications, and external processes can all be defined and initiated as part of MKS Integrity's process automation.

# What Are Item Relationships?

Definable named *relationships* between items are what provide structure to your MKS Integrity configuration, establishing hierarchies and traceability. Items can have an unlimited number of relationships, but standard relationships are defined in the solution templates. For example, a test case relates “upstream” to an originating requirement. Your organization can configure relationships to reflect its own unique needs (and add new relationships as required). Relationships can be used to group items for management and impact analysis as well as driving automated metrics and suspect tracing.

Linking an item to another item creates two relationships: forward and backward. *Forward relationships* are relationships where the related item is a child of the original item. *Backward relationships* are relationships where the related item is a parent of the original item. For example, if you link a master feature to a secondary feature, it creates a forward relationship from the master feature to the secondary feature and a backward relationship from the secondary feature to the master feature. You can create a relationship from either the child or the parent, and the corresponding field in the related item is updated automatically. Every relationship therefore consists of a pair of relationship fields: one for the forward relationship and one for the backward relationship.



*Forward and backward relationships*

## Relationship Field Behavior

Your administrator defines the relationship fields for each item type. The relationship field definition determines the following behavior:

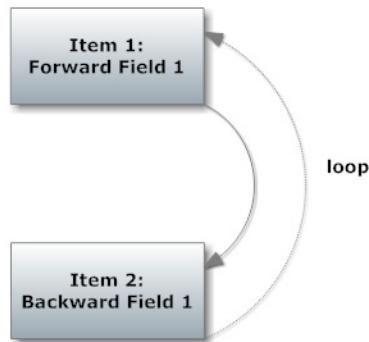
- What types of items can be linked to through the field.
- What panel the relationship field displays.
- The display format of the field (table or comma-separated values).
- Whether multiple items can be linked to through the field.
- Whether the system prevents relationship loops from occurring.
- What relationship flags can be added to item relationships in the field.

## Cycle Detection

*Cycle detection* prevents relationship loops from occurring. A relationship loop occurs when an item has both a forward and a backward relationship with another item within the same relationship hierarchy.

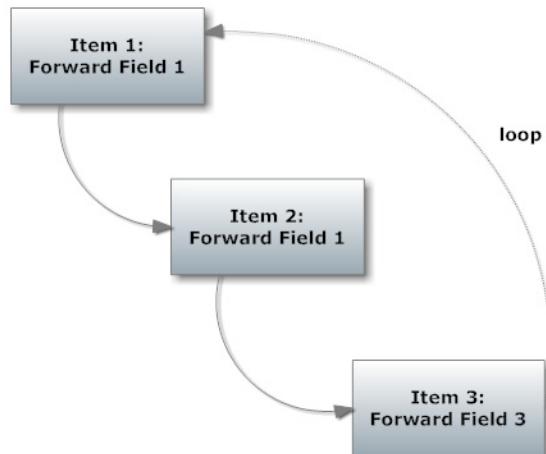
---

For example, if item 1 has a forward relationship with item 2, and you create a forward relationship from item 2 to item 1, you create a relationship loop.



*One level relationship loop*

In another example, if item 3 has a backward relationship to item 1 through item 2, creating a forward relationship from item 3 to item 1 creates a relationship loop.



*Two level relationship loop*

# Items View: Managing Items

## CLI EQUIVALENT `im issues`

The Items view is the central location in MKS Integrity for managing items. Items display in the view based on the results of an item search (by item ID, text, or query). The first time you log in to MKS Integrity, the default Quick Query runs, displaying all items assigned to you.

The Items view is available in the GUI and Web interface, offering most of the same functionality and familiar user interface components.

### To display the Items view

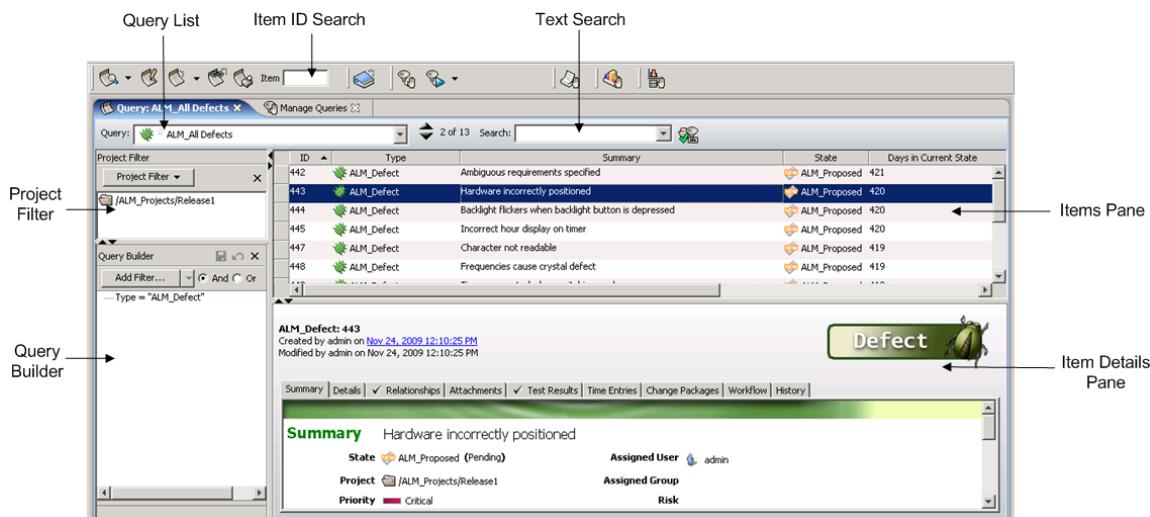
Interface	Procedure
GUI	Select <b>Item &gt; View Items</b> .
Web	Under <b>Quick Views</b> , select <b>Items</b> .

From the Items view, you can perform the following tasks:

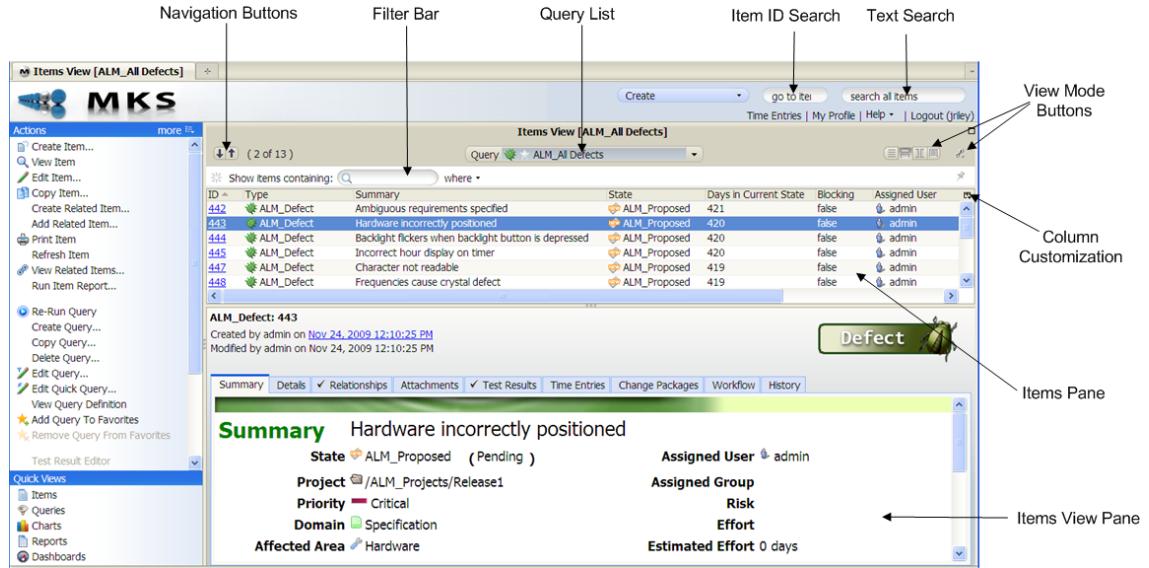
Task	For more information, see...
View the details of an item.	"Viewing Item Details" on page 9.
Search for items.	"Searching for Items" on page 13.
Filter items currently in the view (Web only).	"Filtering Items (Web only)" on page 18.
Create an item.	"Creating an Item" on page 21.
Edit a single item.	"Editing an Item" on page 34
Edit multiple items.	"Batch Editing Items" on page 36
Copy the details of an existing item into a new item.	"Copying an Item" on page 41

### Items View: User Interface Components

In the GUI, the Items view appears similar to the following:



In the Web interface, the Items view appears similar to the following:



The Items view includes the following user interface components:

Component	Description
Item ID Search	<p>Searches for an item by ID. If you know the ID of the item you are searching for, type the ID in the item ID field and press ENTER. For detailed information, see “Searching for Items” on page 13.</p>
Text Search	<p>Searches for items by text. Type the text string you want to search for in the text search field and press ENTER. For detailed information, see “Searching for Items” on page 13.</p>
Query List	<p>Searches for items using a pre-defined query. Click the query list and select a query using the data filter. For detailed information on creating a query, see “Creating a Query” on page 68.</p>
Filter Bar (Web only)	<p>Filters the items currently in the Items pane (in addition to filtering provided by a text search or query). You can also save active filters with the Items view or save them as a query. For detailed information, see “Filtering Items (Web only)” on page 18.</p>
Column Configuration (Web only)	<p>Configures the display of columns in the Items view. Click  and choose a column configuration option. <b>Tip:</b> You can also associate a specific column configuration with a query.</p>
Project Filter (GUI only)	<p>Project filters allow you to query your MKS Integrity database within the context of a project. You can apply a project filter to a query or text search. The project filter uses a data filter to select projects. For more information on using project filters, see “Using Project Filters in the Items View (GUI)” on page 17.</p>
Query Builder (GUI only)	<p>Defines filters for the active query, re-running the query as you refine it. <b>Note:</b> The Query Builder is available in the Web interface, but not as a visible component of the Items view. For detailed information, see “Query Builder” on page 65.</p>
Navigation Buttons (Web only)	<p>Highlights the most recently created, edited, or viewed item in the Items View pane. To highlight a different item, click the arrows.</p>

Component	Description
View Mode Buttons (Web only)	<p>Toggle the display of items and item relationships.</p> <p>To toggle the display of items, click one of the following:</p> <ul style="list-style-type: none"> <li>■ <b>List Mode</b> ( ) displays a list of items in one frame.</li> <li>■ <b>Horizontal Split Mode</b> ( ) displays the list of items in one frame and the details of a highlighted item in the frame below it.</li> <li>■ <b>Vertical Split Mode</b> ( ) displays the list of items in one frame and the details of a highlighted item in the frame beside it.</li> <li>■ <b>Detail Mode</b> ( ) displays the details for a selected item in one frame.</li> </ul> <p>To toggle the display and location of the Relationships view, click  and then one of the following options:</p> <ul style="list-style-type: none"> <li>■ <b>Don't show relationships</b> hides the Relationships view (if visible). By default, the Relationships view is hidden.</li> <li>■ <b>Show in Side bar</b> displays the Relationships view in a frame to the right of the Items pane and/or Item Details pane.</li> <li>■ <b>Show in Footer</b> displays the Relationships view in a frame below the Items pane and/or Item Details pane.</li> <li>■ <b>Show in Detail view</b> displays the Relationships view in a frame below the Item Details pane (only if visible).</li> </ul> <p><b>Show selection in detail pane</b> dynamically updates the Item Details pane and/or Relationships view based on your selection in the Items pane or Relationships view.</p> <p>For detailed information on the Relationships view, see “Relationships View: Viewing Item Relationships” on page 44.</p>
Items Pane	<p>When you perform a global text search or run a query, the results display in the Items pane. You can select an item and display complete item information in the Item Details or Edit Item Details pane.</p>
Item Details Pane	<p>Displays detailed information for the item selected in the Items pane.</p> <p>In the GUI, the Item Details pane is embedded in the Items view by default; however, you can view item details in a single view.</p> <p>For detailed information, see “Viewing Item Details” on page 9.</p>

## Configuring the Items View (GUI)

You can configure the look of the Items view to include what view components display and how the item information displays.

### To configure the Items view in the GUI

- 1 Select **View > Options**. The **Options** dialog box displays.
- 2 On the **General** tab, select the following options as required.

- **View layout**
  - **Show project filter** displays the project filter in the Items view.
  - **Show query builder** displays the query builder in the Items view.
  - **Substitute parameters** displays parameter values for parameter references in text fields in the viewing or editing panel.

---

**NOTE** Parameters are not substituted in the untested items view.

---

- **Variable height rows and inline editing**
  - **Off** displays standard rows in the **Items** pane.
  - **Show variable height rows** displays the entire contents of a field in the Items pane. This allows for images, tables, and other rich content to be displayed.

- 
- **Show variable height rows with inline editing** enables you to edit items directly in the Items pane.
  - **Viewing or editing panel**
    - **Off** displays the Items pane only.
    - **Show item edit view** enables you to edit item details for the item selected in the Items pane. You cannot select this option in combination with the **Show item detail view**.
    - **Show item detail view** enables you to view items details for the item selected in the Items pane. You cannot select this option in combination with the **Show item edit view**.
- 3 If a change to a source field value in a field relationship makes the value in the related target field invalid, the invalid value is cleared automatically. If **Clear values that have been made invalid by a field relationship change** under **Field Relationships** is not enabled, you are prompted with a message and can choose whether to clear the invalid value.
- 4 By default, long field values in tooltips may be truncated. You can display the full value by enabling **Show full field value in tooltip** under **Tooltip Display**.
- 5 On the **Colors** tab, define the colors used to highlight mandatory fields and to highlight the row being edited (when using variable height rows with inline editing).
- 6 On the **Messages** tab, select the following options as required.
- **Show the mandatory field information popup** displays a popup when users have outstanding mandatory fields to complete.

**Mandatory field message** contains the message that displays for outstanding mandatory fields. The default message is: The field {FieldName} is mandatory and requires a value before it can be saved. You can create a custom message to display more specific information, for example, The field {FieldName} is a mandatory field on {Type} : {ID}. Click the **Fields** button to select and insert the fields in the message text.
  - **Show confirmation message before saving a modified item** displays a message when you save a modified item asking you to confirm saving the change.
  - **Show the not editable field information popup** displays a message when attempting to edit a non-editable field.
- 7 On the **Titles** tab, define the layout of the nodes displayed in the Structure column (if visible in the view). The Structure column shows the document structure of the item.
- Enter values in the field surrounded by {} brackets or click the **Fields** button and select the fields from the list.
- You can also enter descriptive text outside the brackets. The text you enter outside of the brackets will be displayed for each item in the Structure column. Adding text can be useful for clarity; for example, a field label.

# Viewing Item Details

---

**CLI EQUIVALENT** im viewissue

---

Once an item is assigned to you or after you locate a specific item, you can view the item details to understand more about the item. For example, you might view item details if you were responsible for reviewing all new Change Request submissions and assigning them to other team members.

Your administrator defines item types, for example, requirements, change requests, defects, or test sessions. Each item type may display a different layout, tabs, fields, and have different names from the default names. In addition, your administrator defines what types and fields are visible to users.

## **Prerequisites**

To view item details, ensure that you have visibility permissions for the item types and fields you want to view.

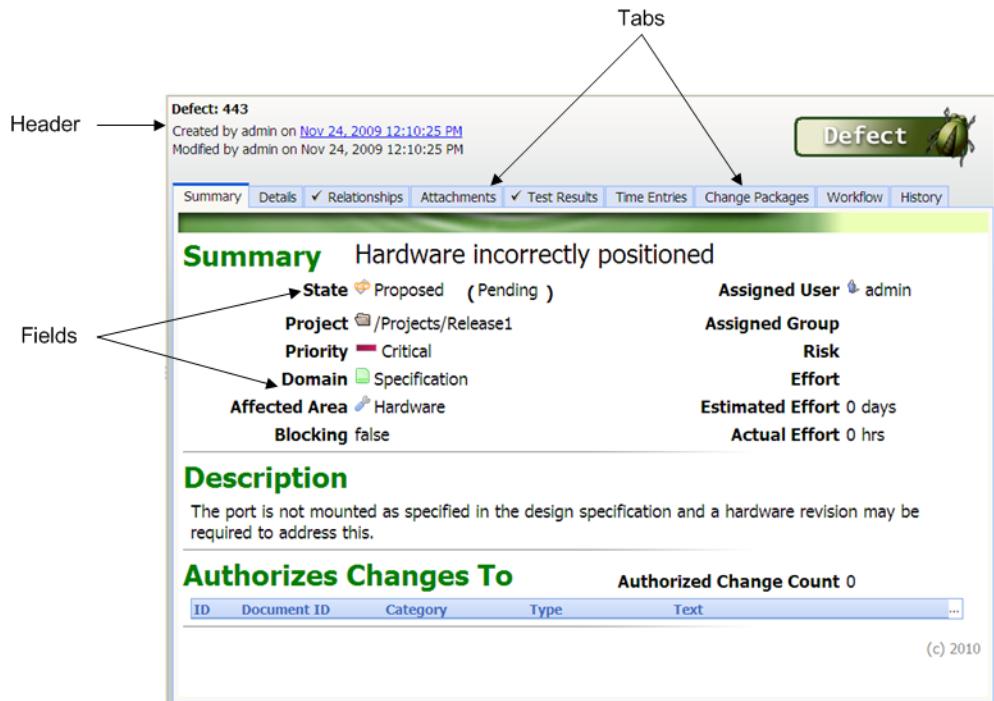
For further assistance, contact your administrator.

## **To view item details**

Interface	Procedure
GUI	<p>In the <b>Items</b> view, select an item and select <b>Item &gt; View Item Details</b>.</p> <p><b>Tip:</b> If you know the ID of an item, you can enter the ID and view the item details by doing one of the following:</p> <ul style="list-style-type: none"><li>■ In the item ID field, type the item ID and press ENTER.</li><li>■ In the Items pane, press CTRL + NUMBER. For example, CTRL 23 selects the item with ID 23.</li></ul>
Web	<p>In the <b>Items</b> view, select an item and select <b>View Item</b>.</p> <p><b>Tip:</b> You can also click the item's ID in the Items view.</p>

## **Item Details: User Interface Components**

The following Item Details view illustrates information in the Defect item type:



The following is an example of the user interface components that display when viewing or editing an item:

Component	Description
Header	<p>The top of the Item Details view displays basic information about the item. By default, the header displays the following:</p> <ul style="list-style-type: none"> <li>■ item type</li> <li>■ ID</li> <li>■ who created the item and when</li> <li>■ who last modified the item and when</li> <li>■ summary of the item</li> </ul> <p><b>Tip:</b></p> <ul style="list-style-type: none"> <li>■ MKS Integrity automatically provides an active e-mail link for users specified in the <b>Created By</b>, <b>Modified By</b>, and <b>Assigned User</b> item fields. Whenever an active e-mail hyperlink displays, you can send an e-mail to that user by clicking the link. If the user does not have an e-mail address, then no link displays.</li> <li>■ The Item Details view displays the entire set of details for a selected current or historical item. Historical items are indicated by the <b>As Of</b> header in the GUI and Web interface which indicates the item's historical context.</li> </ul>

Component	Description
Tabs	<p>By default, tabs organize item data by field type, for example, <b>Fields</b>, <b>Relationships</b>, and <b>Attachments</b>. However, your administrator can edit existing tabs or create new tabs that organize item data in a layout that is relevant to your organization. For example, the <b>Fields</b> tab might be renamed <b>Summary</b>.</p> <p>If the item contains any relationships, attachments, branches, labels, change packages, or time entries,  displays on the respective tab. This allows you to see that the item contains additional information without having to click the tab.</p> <p>To view specific information, click a tab.</p> <hr/> <p><b>Fields</b> (default name) displays item data as entered in predefined fields, for example, <b>Summary</b>, <b>State</b>, <b>Assigned User</b>, <b>Project</b>, <b>Description</b>, or other custom field information.</p> <hr/> <p><b>Relationships</b> (default name) displays related items. For example, a Feature item might have related Documentation and Test Case items.</p> <p>For detailed information, see “Working With Related Item Fields” on page 29.</p> <hr/> <p><b>Attachments</b> (default name) displays a list of attached files with the corresponding name, size, summary, date, and user details.</p> <p>Your administrator defines which item types can have attachments and which item types may require the addition of attachment for a specific state. For example, if you are a developer you may need to include a design document before you can save a Feature item type in the Submit state.</p> <p>For detailed information, see “Working With Attachment Fields” on page 31.</p> <hr/> <p><b>Test Results</b> displays all test results for the item. You can use the data filter to filter the test results by verdict and verdict type.</p> <p>Test results record the outcome of a test case or test step for a test session. Test case, test session, and test step items are special item types defined by your administrator. For a test session item, this tab displays all test results for the session. For a test case item, this tab displays all test results for the test case across all test sessions.</p> <p>Test results can also be related to other item types, as determined by your administrator. For example, a defect item may be related to a failed test result.</p> <p>Test results are designed for use with test management.</p> <hr/> <p><b>Branches</b> displays a list of branches that were created for this item. Branches are created when you begin new, divergent work off of the current or historical version of an item.</p> <hr/> <p><b>Labels</b> displays any labels created for the item. Labels allow you to associate a name with a point in an item’s history; for example, a prior release, project milestone, or document baseline. Double-clicking a label displays the historical item details for the item.</p> <hr/> <p><b>Time Entries</b> displays time spent working on the item by one or more users.</p> <p>For detailed information, see “Time Entries View: Managing Time Entries” on page 56.</p> <hr/> <p><b>Change Packages</b> displays change packages for the item. Change packages group changes to MKS Integrity members that were made as a result of the item. For example, a defect’s change package might include the files that were changed in order to correct a problem.</p> <p>By default, MKS Integrity includes two change package types: MKS Integrity and Implementer. A change package type consists of change package attributes and change package entry attributes.</p> <p>The Web interface also displays the state of Implementer objects in production when they are promoted.</p> <p>For detailed information, see “Working With Item Change Packages” on page 39.</p>

Component	Description
Tabs (continued)	<p><b>Workflow</b> displays complete workflow for item type, unvisited states, visited states, current state, other state transitions, and phases (if enabled) as indicated by <b>Legend</b>. Viewing <b>Workflow</b> panel is useful for determining where you can progress in workflow.</p> <p>In the GUI, you can toggle <b>Legend</b> on or off, and zoom in or out on workflow by right-clicking in <b>Workflow</b> panel and selecting menu item.</p> <p><b>History</b> displays a read-only record of each update made to an item. For example, if you change the item state, the record of change includes your name, date and time of the change, name of the modified field (state), and new value for the field (new state). Since other item values such as project, priority, or assigned user remain the same, they are not included in the change record.</p> <p>Item changes in the <b>History</b> tab indicate the action performed on the item, for example: modify, create, label. Selecting the username link opens a new e-mail message in its native format where the body of the e-mail contains a link to the current item. The date the item was edited displays as a link. Selecting the link opens an Item Detail view for the item as of the date.</p> <p>The <b>History Order</b> options at the top of the panel allows you to configure the order that changes display in. By default, <b>Most recent first</b> is selected.</p> <p>A <b>History list is filtered</b> message displays at the top of the panel if changes made to computed or FVA fields are filtered out of the history. Filtering these fields makes the History tab more usable by removing changes made by MKS Integrity. To turn filtering on and off in the GUI, use <b>View &gt; Show Computed Fields in History</b>. These fields are always filtered in the Web interface.</p> <p>In the GUI, a maximum of 100 changes display. If there are more than 100 changes, the current page and the number of pages display. In addition, navigation buttons display the next, previous, first, and last page.</p> <p>To improve performance in the GUI, attachment images do not display in the <b>History</b> tab. A <b>Display All Images</b> button displays beside a placeholder image. Clicking the button displays the actual attachment image. The actual image displays until you close the Item Details view.</p>
Fields	<p>Fields display item data as entered by users or based on rules defined by your administrator. For example, you may be able to define information for the <b>Summary</b>, <b>State</b>, <b>Assigned User</b>, <b>Project</b>, and <b>Description</b> fields, or some of these fields may automatically fill out based on a state transition.</p> <p>Your administrator defines what fields display in an item type, who can view and edit them, and the names of the fields.</p>

---

# Searching for Items

By default, the Quick Query is available to all users and searches for all items assigned to you. However, you can search for more specific items in the following ways:

- **Search by Item ID**

In the item ID field, type the item ID and press ENTER.

---

**TIP**

- In the GUI, click the drop-down arrow beside the item ID field to display a selectable list of the 10 most recently typed IDs.
  - In the Web, click the arrow beside the item ID **go to item** field to display the most recent ID search, allowing you to refine the search.
  - When you search for an item, the **Item ID not found** message displays when the item does not exist and when you do not have permission to view the item.
- 

- **Search by Text**

In the text search field, type the text string you want to search for and press ENTER.

---

**TIP**

- The text search function uses a search syntax similar to many common Web search engines; however, it can also use a syntax specific to your database type. For more information, see “Enhancing a Text Search for Items Using Operators” on page 13.
  - In the GUI, click the drop-down arrow beside the text search field to display a selectable list of the 10 most recently typed text strings.
  - In the Web, click the arrow beside the text search field to display the most recent text search, allowing you to refine the search.
- 

- **Search Using a Query**

Click the query list and select a query using the data filter.

For detailed information on creating a query, see “Creating a Query” on page 68.

## Enhancing a Text Search for Items Using Operators

The search function in MKS Integrity allows you to carry out simple text searches of your item database.

The text search function uses a search syntax similar to many common Web search engines. You can also perform searches using the search engine of the underlying database, and the results are passed back to MKS Integrity.

### ***Filtering by Project***

In the GUI, you can also apply project filters when you conduct a text search. For example, you could search for items within the `SourceCode` project that contained the string “`null pointer exception`”.

In the Web UI you can use filters to display items from a specific project or projects (for more information see “Filtering Items (Web only)” on page 18), and then perform a text search on the results.

---

## Saving Text Searches

When you run a text search, that search becomes available in the query list. In the GUI, you can save this text search as a query by copying it. In the Web interface, you can save this text search as a query by right-clicking **where** in the filter panel and select **Save Filters As New Query**. You can then perform further modifications if required. For more information on copying a query, see “Copying a Query” on page 79. For more information on saving a filter as a query, see “Filtering Items (Web only)” on page 18.

---

**NOTE** Once you run another query, the text search query item is removed from the query list.

---

## Text Search Operators

The search feature supports full text searching using certain operators to enhance the search.

---

**NOTE** Your administrator may decide to configure MKS Integrity without full text searching. If text searching is not enabled, the search options listed in this section are not valid.

---

The following table provides general information for carrying out text searches of supported databases:

Search Operator	Description
AND	Most databases ignore common words such as “a” or “the” in search text. Using AND in front of a common word allows you to include it in your search. You can also use the + symbol to represent the AND operator. Another way of including common words in a search is to search for a phrase in quotation marks (see “ ” below).
OR	Allows you to search for text strings that contain either of the specified words. You can also use the   symbol to represent the OR operator. Using multiple OR operators can increase the time required for your text search.
-	Allows you to exclude a word from your search by using the – symbol in front of it. Make sure you include a space before the – symbol. Using this operator causes your search to be applied against each text field separately. This means that if a text field in an item meets the search criteria, the item displays in the search results. For example, if an item contains two text fields: field1="x" and field2="y", and you search on "x -y", the item displays in the search results because field1 meets the search criteria.
~	Allows a fuzzy search for words or phrases similar to the specified text string.
*	A wildcard allowing you to search for the specified string with any other text. How to use the wildcard depends on the underlying database. For example, SQL only allows the * to be placed as a suffix while Oracle allows the * to be placed as a prefix or suffix to the word.
LIKE	Allows you to search for words that are similar to the specified string. Using this operator causes your search to be performed without using indexes, which will result in your search taking longer.
DB	Allows you to search for text using the search engine of the underlying database.

## Searching DB2

You can perform a text search on a DB2 database by putting DB in front of your text search string. The format of the text search string is defined by the IBM DB2 Net Search Extender. To search MKS Integrity items stored in a DB2 database, use:

- double quotes ("words") for each word you are searching for
- & between two words to search for both words
- | between two words to search for either word

- 
- NOT to search for words that are not present
  - FUZZY FORM OF "word" to search for words that are similar to the search term
  - closed brackets () to group terms together

For example, the text search string:

```
DB "database" & FUZZY FORM OF "exception" & ("new" | "old") & NOT "test"
```

performs a search for text containing the term database, with the word exception possibly misspelled, one of the words new or old, but not containing the term test.

For rich content fields, a text search returns results from text in rich content fields only. You cannot search for HTML elements or attributes in rich content fields. For more information on rich content fields, see "Working With Rich Content Fields" on page 24.

For more detailed information and advanced usage, refer to your IBM DB2 documentation.

### **Searching Oracle**

You can perform a text search on an Oracle database by putting DB in front of your text search string. The format of the text search string is defined by the Oracle Text Package. To search MKS Integrity items stored in an Oracle database, use:

- AND between two words to search for both words
- OR between two words to search for either word
- NOT to search for words that are not present
- FUZZY(word) to search for words that are similar to the search term
- closed brackets () to group terms together
- \character to escape a character

For example, the text search string:

```
DB database AND FUZZY(exception) AND (new OR old) NOT test
```

performs a search for text containing the term database, with the word exception possibly misspelled, one of the words new or old, but not containing the term test.

For rich content fields, a text search returns results from text in rich content fields only. You cannot search for HTML elements or attributes in rich content fields. For more information on rich content fields, see "Working With Rich Content Fields" on page 24.

For more detailed information and advanced usage, refer to your Oracle Database documentation.

### **Searching MS SQL**

You can perform a text search on an MS SQL database by putting DB in front of your text search string. The format of the text search string is defined by Microsoft SQL Server Full Text Support. To search MKS Integrity items stored in an MS SQL database, use:

- double quotes ("phrase") around a phrase to search for several words
- AND between two words or phrases to search for both words or phrases
- OR between two words or phrases to search for either word or phrase
- AND NOT to search for words that are not present
- <prefix>\* to search for words starting with a prefix
- closed brackets () to group terms together

- 
- <word> NEAR <word> to search for words or phrases in close proximity

For example, the text search string:

```
DB (database NEAR exception) AND (new OR old) AND NOT test
```

performs a search for text containing the term database near the word exception, with one of the words new or old, but not containing the term test.

For rich content fields, a text search returns results from text in rich content fields and the underlying HTML elements and attributes. For more information on rich content fields, see “Working With Rich Content Fields” on page 24.

For more detailed information and advanced usage, refer to your MS SQL Server documentation.

### **Key Considerations**

- Text searches look only for information that is in short or long text fields. The search does not capture information in other types of fields, such as integer, pick, floating point, logical, date, user, or group. Symbols such as !, #, and @ are ignored in the search. If there is an error in the search, no items are returned.
- If you run a global text search in the GUI and the Items view is the active view, the search results display in the active Items view.  
If you run a global text search in the GUI and a non-Items view is the active view, the search results display in a new Items view.  
If you run a global text search in the GUI and an Items view is not open, the search results display in a new Items view.
- In the GUI, holding down **CTRL** while pressing **ENTER** opens the search results in a new tabbed Items view.

### **To search for text using a project filter (GUI only)**

- 1 In the **Items** view, click  beside the **Search** field on the toolbar. The project filter changes to  indicating project filters are enabled.
- 2 Type the text string you want to search for.
- 3 Choose the project(s) you want to search from the directory tree in the **Project Filter** pane.

---

**NOTE** To clear all project selections, click **Unspecified** and then clear that highlight using **CTRL+click**.

---

Project filtering is enabled and the selected text search is filtered according to the project(s) you have selected.

---

# Using Project Filters in the Items View (GUI)

A *project filter* allows querying within the context of a selected project.

A project filter enables you to apply a project filter on top of a query. This enables you to narrow down query results when searching for an item or items without changing the query. It also enables you to maintain a set of commonly used projects, which you can then filter further as required.

By default, project filtering is enabled but with no projects selected and therefore displays items in *all* projects.

You can also enable project filtering for text searches. For more information, see “To search for text using a project filter (GUI only)” on page 16.

---

**NOTE** Selecting **Unspecified** means MKS Integrity searches to find items where no project is assigned.

---

## Example

In his Quick Query, Chad sees all his items, but he is only interested in items that belong to the ABC Tools project. Chad uses the project filter to display only ABC Tools project items.

## To use project filtering in the GUI

- 1 To enable project filtering, select **View > Options**, then select **Enable Project Filter**. The **Project Filter** pane displays.
- 2 Click **Project Filter** and use the data filter to select the projects that you usually work with.
- 3 The selected projects display in the **Project Filter** pane and the **Items** view is refreshed to only show items that belong to the displayed projects.
- 4 If you want to filter further within the projects in the **Project Filter**, select one or more of the displayed projects. The **Items** view is refreshed to only show items that belong to the selected projects.

---

**NOTE**

- Selecting all the projects is the same as selecting none of them: it shows items that belong to all the displayed projects.
  - To cancel the selection of one or more projects in the project filter, hold down the **CTRL** button and left click the project selection(s) that you want to cancel.
- 

The last project filter used is saved with the Items view and runs the next time you open the view. Any additional filtering is not saved with the view.

# Filtering Items (Web only)

After you run a query to search for items, you can apply additional filtering to the Items view in the Web interface to further refine the query results. At the top of the view, a filter bar allows you to filter the items currently in the view by text and/or using pre-defined filters.

The filter criteria creates a sentence in the filter bar that indicates what you are looking for, for example: Show items containing Financial Calculator where Assigned User is one of "John Riley (jriley)" Project is one of "Release 2.0".

Once you add filters, you can manage them by doing the following:

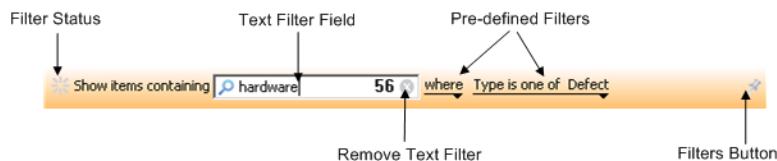
- **Pin (apply) a filter to all queries**

You can pin (apply) one or all filters to all queries. *Pinning* a filter saves the filter with the current view and subsequent views you open so that the filters are always used, even when different queries are run. For queries, pinned filters are used in addition to the query filters.

- **Save the current filters as a new query**

Once you define a set of filters, you can save them along with your existing query as a new query for easy re-use.

The following is an example of an active filter bar in a view:



The filter bar includes the following user interface components that allow you to refine query results:

Component	Description
Filter Status	As you apply filters, the filter status icon animates to indicate progress.
Text Filter Field	Filters items by text, based on text in the visible columns in the Items view.
Remove Text Filter	Clears the text filter field. It does not remove any enabled pre-defined filters.
Pre-defined Filters	Pre-defined filters from the <b>where</b> list enable you to filter based on values in the visible columns in the view. These filters are cleared when you run a different query. These filters are cleared when you click the <b>where</b> filter and select <b>Reset</b> .
Filters Button	Manages pinned and saved filters. As you work with filters, the Filters Button changes to indicate the status of the filter(s): If there are no pinned filters, the filters button is gray ( ). If there are pinned filters, the number of filters displays beside the filters button, for example, . If the pinned filters are disabled, the button is gray with the number of filters beside the filters button, for example, .

## Filter Bar Tasks

From the filter bar, you can perform the following tasks:

Task	Procedure
To filter items based on text	Type the text that you want to filter by. The list of items is filtered immediately as you type. To clear the text, click  at the end of the text filter field.
To add a filter based on the fields in the visible columns	Click <b>where</b> and select a field, for example, <b>State</b> . Define the filter criteria for the selected field.
To add a filter	Click <b>where</b> and select <b>Add Filter (Web)</b> . Define the filter criteria.
To select a recently run filter	Click <b>where</b> , select <b>Recent Filters</b> , and select the filter you want to use from the menu.
To edit a filter	Click the filter component that you want to edit and select the configuration you want to change from the menu. Change the filter criteria as required.
To invert a filter	Click the filter component that you want to invert, then select the inverted value. For example, if the filter is <b>Type is "Defect"</b> , the inverted value is <b>Type is not "Defect"</b> .
To remove a filter	Click the filter component that you want to remove, then select <b>Remove Filter (Web)</b> from the menu.
To remove all filters	Click <b>where</b> and select <b>Reset</b> .
To save a filter as a query	Click <b>where</b> and select <b>Save Filters As New Query</b> . Enter a query name. You have the option of saving applied filters as part of the query. <b>Note:</b> This command creates a new query by saving the existing query along with the filters you have defined. The original query is included in the new query.
To pin a filter to all new queries	Click the filter component that you want to apply, then select <b>Apply Filter to All New Queries</b> from the menu. <b>Note:</b> The last filter you pin to a view is pinned to each subsequent view you open. For example, if you pin a filter to an <b>Items</b> view, the filter is also applied to the next <b>Items</b> view you open. The filter is not pinned to other currently open views, or views of a different type, for example, a <b>Relationships</b> view.
To pin filters to all queries	Click <b>where</b> and select <b>Apply Filters to All Queries</b> .
To view pinned filters that apply to all queries	Click the pinned filter.
To modify a pinned filter	Click the pinned filter, select the applied filter that you want to modify, and select the modification option in the sub menu. <b>Note:</b> When you close and restart the Web, the last modified pinned filter is applied to all views containing a pin.
To remove a pinned filter	Click the pinned filter, select the applied filter that you want to remove, and click <b>Remove Filter</b> .
To pin a filter to the current query only	Click the pinned filter, select the applied filter that you want to remove, and click <b>Unpin Filter from this Query (Web)</b> .
To remove all pinned filters	Click the pinned filter and select <b>Remove Filters</b> .
To temporarily disable pinned filters	Click the pinned filter and select <b>Disable Filters</b> .
To re-enable disabled pinned filters	Click the disabled pinned filter and select <b>Disable Filters</b> .

## Using the Filter Editor Dialog Box (Web only)

Depending on the type of filter you are adding or editing, not all of the following steps apply.

- Select the field you want to filter on. For example, you could select an integer field such as **ID**, or a date field such as **Created Date**. After selecting the field, click **Next**.

- 
- 2 Select the operator for the field filter (if required). For example, for an integer field, you could select **is greater than (>) a number**; for a date field, you could select **is in the last few months**. Click **Next**.
  - 3 Select the value(s) for the field or enter the value for the operator. For example, for a filter criteria of **is greater than (>) a number**, enter the number that the field value must be greater than; for a filter criteria of **is in the last few months** enter the number of months in the past that the date field can contain a date for; for a field value for the **Assigned User** field, select the assigned user value(s).

When entering a value for an operator,  displays beside the field until you enter a value. When you enter a valid value,  displays.
  - 4 To save the filter, click **Finish**.

---

# Creating an Item

---

**CLI EQUIVALENT** im createissue

---

You create an item by providing item data. You can also link the item you are creating to another item or add an attachment to it.

By default, fields that you can modify display on the following three tabbed panels: **Fields**, **Relationships**, and **Attachments**. However, depending on how your administrator has defined the item type, these tabs might have different names or may not display. Your administrator may also define additional custom tabs to display fields, attachments, and relationships.

## **Prerequisites**

Before you create an item, ensure that you have the permissions to create the item type and fill out the relevant fields.

If there are no item types to select or if you can select an item type but not create it, contact your administrator. The item may contain a custom field defined to be both mandatory and invisible for this item type.

## **Key Considerations**

- If your administrator has set up electronic signatures, you may need to provide your user name and password when creating an item.
- Relevance and editability rules are evaluated based on the MKS Integrity Client's time zone.
- For detailed information on working with the different types of fields that may appear in an item, see "Rich Content: Key Considerations" on page 25.

---

## To create an item

Task	Procedure
Create an item	<p>In the GUI, select <b>Item &gt; Create</b>.</p> <p>In the Web interface's Items view, select <b>Create Item</b>.</p> <p>Select an item type and click <b>OK</b>.</p> <p>Fill out the necessary fields in the item. As soon as you add information, an asterisk (*) displays in the title bar, indicating there is unsaved information in the item. This information is not committed to the database until you click <b>Save &amp; Close</b> or <b>Save</b>. Once you click <b>Save &amp; Close</b> or <b>Save</b>, the status bar indicates that the information is being committed to the database, and an ID is assigned to the item.</p> <p><b>Tip:</b></p> <ul style="list-style-type: none"><li>■ In the GUI, you can also create an item when editing items inline from the Items and Relationships views. For more information on inline creation of items from the Items view, see "Editing Items Inline (GUI Only)" on page 33.</li><li>■ You can also select <b>Create item type</b>.</li><li>■ Press <b>INSERT</b> to create an item in the GUI or Web interface.</li><li>■ In the Web interface, you can also click the <b>Create</b> field in the title pane and select the item type to create.</li></ul>
Commit the item information to the database and resume entering additional information	<p>Click <b>Save</b>. The status bar indicates that the information is being committed to the database, and the <b>New/Create item</b> view becomes an <b>Edit item</b> view.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"><li>■ Each <b>Save</b> operation is added to the item history.</li><li>■ When you commit item information to the database, fields in the item dynamically update based on relevance and editability rules. This means that what you can see and do in the item may change.</li></ul>

## Working With Field Types

The following describes working with the different types of fields that may appear in an item:

Field Type	Notes
Text	<ul style="list-style-type: none"><li>■ Your administrator can configure long text fields to support <i>rich content</i>. Rich content enhances the display of text in long text fields by adding formatted text, tables, background colors, images, and hyperlinks. For more information on rich content, see "Working With Rich Content Fields" on page 24.</li><li>■ Text fields, such as <b>Summary</b>, support HTTP hyperlinks. This is useful if you want to include a link to a document on an internal or external Web site, such as a design document. You must include the <code>http://</code> prefix.</li><li>■ You can specify a parameter name in short or long text fields. For more information, see "Editing an Item" on page 34.</li></ul>
Users and Groups	<ul style="list-style-type: none"><li>■ The users that can be selected for user fields are limited to those with permissions to the item's project. The same applies to group fields.</li><li>■ Inactive users and groups are not available for selection. If a selected user or group value becomes inactive and the field remains untouched during an item edit, the inactive value persists. If you modify the field with the inactive value, you must change it to an active value.</li></ul>
Attachments	<ul style="list-style-type: none"><li>■ If mandatory attachments were defined for a specific state in your workflow, you must add an attachment before you save the item and a state transition can occur. For example, a developer may be required to attach a design document to a <b>Feature type</b> in the <b>In Design</b> state.</li><li>■ Attachment fields can display in comma-separated values (CSV) or table display format, depending on how your administrator has defined the attachment field.</li><li>■ By default, MKS Integrity allows file attachments to a maximum size of 4 MB and a maximum of 255 characters for file names. Your administrator may define a higher or lower limit depending on the requirements of your system. You can also attach more than one file to a single item.</li></ul>

Field Type	Notes
Relationships	<ul style="list-style-type: none"> <li>■ Relationship fields can display in comma-separated values or table display format, depending on how your administrator has defined the relationship field.</li> <li>■ Adding a related item is only permitted if your administrator has allowed relationships for the item type.</li> <li>■ You can create trace relationships between two items in the document model repository; for example, between a requirement and a test case, or between a higher-level requirement and a lower-level requirement. Trace relationships are defined through field pairs and are presented to the user in domain-specific language, for example, test management or requirements.</li> </ul>
Workflows and Document Projects	<p>Inactive projects are not available for selection. If a selected project value becomes inactive and the field remains untouched during an item edit, the inactive value persists. If you modify the field with the inactive value, you must change it to an active value.</p>
Configuration Management Projects	<p>To retrieve metrics from a configuration management project related to the item you are creating, your administrator may define a field that accepts a configuration management project as a value. Optionally, you can specify a checkpoint or development path. If you specify an project checkpoint, then save the item, one or both of the following may occur:</p> <ul style="list-style-type: none"> <li>■ One or more computed expressions in the item calculate specific metrics about the project, displaying the results as a read-only value in a computed field (the visibility of the computed field depends on the field's relevance rules). For example, once you specify a project for the <b>Source Code</b> field, a <b>Lines of Code</b> field could calculate and display the number of lines of code in that project. As lines of code are added or removed from the project, the <b>Lines of Code</b> field updates to display the new value.</li> <li>■ A <b>metrics</b> hyperlink displays in the configuration management project field. Clicking the hyperlink displays various metrics about the project.</li> </ul> <p>In addition, the server and project information display in the configuration management project field as a hyperlink. Clicking the hyperlink displays the project in a <b>Project</b> view.</p> <p>In MKS Integrity, the item displays on the <b>Associated Items</b> tab for the specified project.</p> <p>To select a configuration management project, you require the <b>OpenProject</b> permission for the specified project. Once a project has been specified, metrics can be obtained by any user with permissions to view the project field.</p> <p><b>Important:</b> Metrics are only maintained against project checkpoints; therefore, to generate metrics, you must specify a checkpoint when you specify the project.</p>
Numeric	<ul style="list-style-type: none"> <li>■ To quantify numeric field values, your administrator may define display patterns for numeric fields, for example, as currency or percentages. In addition, a display pattern may format the value you initially enter in the numeric field; for example, an input value of 0.126 may display as 0.13 after you create the item.</li> <li>■ If a floating point field does not contain a display pattern, the field in the created item displays the same number of decimal places as when the value was typed in the numeric field.</li> <li>■ Integer fields allow a maximum of nine digits and floating point fields allow a maximum of 15 digits. Your administrator can define default, minimum, and maximum values.</li> </ul>

Field Type	Notes
Date	<ul style="list-style-type: none"> <li>■ Your administrator may include the time in date fields. You can specify the time when you select a date from the calendar. Time is specified in 24 hour format from 00:00:00 to 23:59:59 inclusive; however, MKS Integrity displays the time in 12 hour format. For example, specifying 13 : 56 : 45 displays the time as 1 : 56 : 45 PM. If you do not specify a time, the current time displays in the date field.</li> <li>■ If you cannot clear a date field, clicking <b>Clear Date</b> enters the current date.</li> <li>■ Dates/times can only be changed through the calendar; they cannot be specified in the field.</li> <li>■ The resulting date/time is reflected in the corresponding date/time field using the server's locale sensitive date/time display format. Time zone is not displayed, implying the client's time zone.</li> <li>■ Displayed date fields do not change based on the time zone a user operates in; however, displayed date/time fields do vary based on the time zone a user operates in.</li> <li>■ Months and days of the week in the calendar popup are locale sensitive. They are based on the server's locale.</li> <li>■ Time zone information displays in the calendar popup using the client's current time zone, for example, GMT-5:00. The time zone display does not change when the selected date/time crosses the daylight savings boundary, though it is recorded with the correct time zone.</li> <li>■ Computed expressions return dates/times in the MKS Integrity Client's time zone and perform calculations in the MKS Integrity Server's time zone where appropriate.</li> <li>■ When entering the date (or time) using the keyboard, it must be typed in the format as displayed in the field. Date fields include the following navigational keyboard shortcuts for accessibility and ease of use: <ul style="list-style-type: none"> <li>■ F4 opens the calendar for selection.</li> <li>■ SHIFT + TAB moves focus of selection.</li> <li>■ ALT+C presses the <b>Clear Date</b> button.</li> <li>■ ALT+Y presses the <b>Today</b> button (if exists).</li> <li>■ ALT+N presses the <b>Now</b> button (if exists).</li> <li>■ Left, right, down, and up arrow keys navigate the days, months, and years (or hours, minutes, and seconds).</li> <li>■ PAGE UP and PAGE DOWN navigates the days, months, and years.</li> </ul> </li> </ul>
Phase	Your administrator may define phase fields in your workflow. Phase fields are read-only fields that specify categorized groups of states in a workflow, essentially creating states (represented by phases) and substates (represented by states) for an item type. Phases are useful for organizing an item type's workflow if it contains a large number of states and to provide users with a broad overview of an item's status independent of the workflow. Your administrator may display phases in the <b>Workflow</b> tab, if enabled.
Parameter and Parameter Value	Your administrator may define parameter and parameter value fields for item types.

## Working With Rich Content Fields

Rich content enhances the display of text in long text fields by allowing you to add formatted text, tables, background colors, images, and hyperlinks (to images, websites, and items). Your administrator enables rich content in long text fields and can define Cascading Style Sheets (CSS) to ensure a consistent look when viewing and printing rich content field information in different Web browsers.

Rich content is expressed using a limited set of HTML elements and attributes. In the GUI and Web interface, rich content is applied using menu commands, shortcut menu commands, and toolbar buttons. You can also add rich content by copying and pasting HTML source into a rich content field. For more information on working with HTML elements and attributes, see “Rich Content: HTML Elements and Attributes” on page 27.

---

### **Prerequisites**

In the GUI, the rich content menus and toolbar buttons must be enabled in the active ViewSet. Once enabled, rich content functionality displays in menus and toolbar buttons when you select a rich content field while creating, editing, or copying an item.

In the Web interface, rich content functionality is available as a toolbar and does not need to be enabled. When you create, edit, or copy an item, selecting a rich content field displays all rich content functionality in a rich content toolbar.

### **To apply rich content in a long text field**

Select the rich content field. Rich content functionality becomes available from the **Format** and **Edit** menus (GUI only) and toolbar buttons (GUI and Web interface). For details on working with specific types of rich content, see “Rich Content: Key Considerations” on page 25.

### ***Rich Content: Key Considerations***

When working with rich content, consider the following:

---

Rich Content Area	Notes
General	<ul style="list-style-type: none"> <li>■ You cannot copy and paste HTML source that exceeds the maximum allowed characters for the rich content field. If you exceed the amount, MKS Integrity allows you to attempt the operation again until you copy and paste an acceptable amount of characters.</li> <li>■ <b>CTRL+SPACE</b> inserts a non-breaking space at the insertion point.</li> <li>■ To delete content in a list, select the content or place the cursor after the content in the list and then press the <b>DELETE</b> or <b>BACKSPACE</b> keys. If the cursor is on the row following a list, pressing the <b>DELETE</b> or <b>BACKSPACE</b> keys does not delete any content in the list.</li> <li>■ The cut, copy, and paste commands are only available in the GUI. In the Web interface, use shortcut keys or the browser's cut, copy, and paste toolbar buttons.</li> <li>■ Hyperlinks in the GUI must contain <b>HTTP://</b> or <b>HTTPS://</b> to function in rich content fields. Similarly, hyperlinks in the Web interface are only navigated correctly when viewing an item if they begin with <b>HTTP://</b> or <b>HTTPS://</b>. When an item is edited, other paths for hyperlinks may navigate correctly but the behavior is dependent on the browser used.</li> <li>■ To follow a hyperlink in the field, press <b>CTRL + click</b> the hyperlink.</li> </ul>
Tables	<ul style="list-style-type: none"> <li>■ In the Web interface, table commands are available when you right-click a table. In the Firefox browser, clicking the arrows in a table allow you to manipulate the table.</li> <li>■ If rich content field information cannot be displayed properly in a table, it displays as plain text.</li> <li>■ When selecting table elements using the mouse: Selecting the contents in a cell does not select the entire cell. However, if the selection continues into an adjacent cell, both cells are selected. <b>CTRL +clicking</b> anywhere inside a table cell selects that cell. Single-clicking anywhere inside a table does not select anything. Double-clicking on the first valid position in a cell or anywhere else inside the cell selects the text at the click position. However, double-clicking at the end of a cell does not select anything in the cell. Triple-clicking on the first valid position in a cell, end of a cell, or anywhere else inside the cell selects the entire cell.</li> <li>■ When navigating in table cells using keyboard shortcuts: If the cursor is in the first cell, <b>TAB</b> moves the cursor to the beginning of the second cell. If the cursor is in the second cell, <b>TAB</b> moves the cursor to the beginning of the third cell. If the cursor is in the second cell, <b>SHIFT+TAB</b> moves the cursor to the beginning of the first cell. If the cursor is in the last cell, <b>TAB</b> creates a new row If the cursor is in the last cell, <b>SHIFT+TAB</b> moves the cursor to the beginning of the second-last cell.</li> <li>■ When copying and pasting table cells, you can select the table row or a single cell.</li> <li>■ When copying and pasting a table row, the pasted row is inserted as a new row above the currently selected row.</li> </ul>

---

Rich Content Area	Notes
Attachments and Images	<ul style="list-style-type: none"> <li>■ To improve performance in the GUI, attachment images do not display in the <b>History</b> tab.</li> <li>■ Inserted images can be resized or moved by dragging.</li> <li>■ If you insert an image referenced in another item, clicking the image opens the item containing the image.</li> <li>■ Double-clicking an inserted image or attachment opens the <b>Edit Attachment</b> dialog box, allowing you to make changes.</li> <li>■ The undo tool does not apply to inserted images or attachments.</li> <li>■ When copying and pasting HTML source that includes images, only images in external Web sites are pasted.</li> <li>■ Pasted images from HTML source that is not an external Web site are assigned the following naming convention: MKSImg&lt;#&gt;.png.</li> <li>■ When inserting an image or an attachment, MKS Integrity displays the attachment fields you are allowed to use for the rich content field. Attachment fields are affected by visibility and type restrictions.</li> <li>■ In the Web interface, you can copy and paste an image containing a local file URL (<code>file://</code>); however, MKS Integrity warns you that the image will not be visible to other users because local file URLs to images cannot be saved as attachments.</li> <li>■ If you insert an image file as an attachment in the Web interface, the image file displays as the pending attachment icon (  ) until you save the item. After you save the item, the icon displays as the image you added.</li> <li>■ URL links to files on local file systems are converted to attachments; however, files from remote file systems are not converted to attachments.</li> <li>■ Inserted attachment file names (including images) containing an ampersand (&amp;) are stored as %26 instead of &amp; by MKS Integrity, even if the attachments are added as a URL or pasted.</li> <li>■ Copying and pasting inline image attachments: from one item to another results in a new attachment associated with the destination item within the same item reuses the attachment in each location when batch editing multiple items creates a new attachment associated with all items, regardless of where the original attachment was copied from</li> </ul>
HTML Elements and Attributes	<ul style="list-style-type: none"> <li>■ Detailed information about HTML is beyond the scope of this guide. To copy and paste HTML source or type HTML from the CLI, MKS recommends experience with HTML.</li> <li>■ Typing HTML tags in a rich content field from the GUI or Web interface does not apply the HTML tags. For example, typing <code>&lt;b&gt;text&lt;/b&gt;</code> displays <code>&lt;b&gt;text&lt;/b&gt;</code> when you save the item.</li> <li>■ In the GUI, unsupported HTML elements and attributes are silently discarded when you paste HTML source into a rich content field. In the Web interface, unsupported HTML elements and attributes are silently discarded after you save the item.</li> </ul>

### ***Rich Content: HTML Elements and Attributes***

Rich content fields support the following HTML elements and attributes:

Element	Attributes
<!-- MKS HTML -->	Mandatory element that prefacing all elements in a rich content field. Other comments are removed.
<h1> to <h6>	align
<p>	style, align
<hr>	size, width
<ol>, <ul>, <li>	style

Element	Attributes
<img>	src, border, alt, align, hspace, vspace, height, width, style
<a>	href, title, onclick For item links, href is: mks:///item?itemid=itemID where itemID is the ID of the item you want to link to. For attachment links, href is: mks:///item/ field?fieldid=AttachmentField&attachmentname=attachment where AttachmentField is the name of the attachment field containing the attachment and attachment is the name of the attachment you want to link to.
<span>	style
<font>	style, color
<div>	style, align
<blockquote>	
<strong>	Converted to <b>.
<b>	
<i>	
<code>	
<em>	Converted to <i>.
<strike>	
<del>	Converted to <strike>.
<pre>	
<table>	style, align, border, width, cellpadding, cellspacing, summary, frame, rules, bgcolor, background
<tbody>	
<tr>	style
<td>	style, rowspace, align, height, width, bgcolor, valign, background
<caption>	

### Item Rich Content: Insert <Object> Dialog Box

The **Insert <Object>** dialog box specifies options for the object you want to insert into the rich content field.

Object	Field	Description
Image	From	Specifies the origin of the image: <ul style="list-style-type: none"> <li>■ Attachment inserts an image from an attachment field in the current item. Click <b>Browse</b> to locate the image.</li> <li>■ URL inserts an image from a URL you type or by clicking <b>Browse</b> to locate the image.</li> </ul>
Attachment Link	From	Specifies an attachment from an attachment field in the current item. Click <b>Browse</b> to locate the image.
	Label Type	Specifies the label to apply to the attachment link: <ul style="list-style-type: none"> <li>■ Text specifies a label you type.</li> <li>■ Attachment specifies an attachment from an attachment field in the current item. Click <b>Browse</b> to locate the image.</li> <li>■ Link inserts an attachment from a URL you type or by clicking <b>Browse</b> to locate the attachment.</li> </ul>

Object	Field	Description
Hyperlink	Link	Specifies the URL of a hyperlink.
	Label	<p>Specifies the label to apply to the hyperlink:</p> <ul style="list-style-type: none"> <li>■ Text specifies a label you type.</li> <li>■ Attachment Image specifies an attachment from an attachment field in the current item. Click <b>Browse</b> to locate the image.</li> <li>■ URL Image specifies a URL to an image you type or by clicking <b>Browse</b> to locate the image.</li> </ul>
Item Link	Item	Specifies the ID of an item you type or by clicking <b>Browse</b> to locate the item.
	Label	<p>Specifies the label to apply to the item link:</p> <ul style="list-style-type: none"> <li>■ Text specifies a label you type.</li> <li>■ Attachment Image specifies an attachment from an attachment field in the current item. Click <b>Browse</b> to locate the image.</li> <li>■ URL Image specifies a URL to an image you type or by clicking <b>Browse</b> to locate the image.</li> </ul>

## Working With Related Item Fields

By default, related items display on the **Relationships** tab. However, they may display on other tabs depending on how your administrator has defined the item type.

### Prerequisites

Depending on how your administrator has defined the relationship field, you may be restricted to only one related item.

### Key Considerations

- You can open related items and edit them, but the changes do not display in the related items field until you refresh the field by pressing F5.
- You can right-click to customize the columns in a relationship field.
- If your ViewSet contains both the Items view and the Item Detail view, you can add a related item using drag-and-drop. Display the item you want to create a relationship for in the Item Detail view, click the item you want to add as a related item in the Items view, and drag it to the relationship field on the target item.
- In the Items view, you can add a related item using drag-and-drop.

Click the item you want to add as a related item in the Items view, and drag it to the target item. An arrow icon and As Relationship tooltip display.

Release the mouse button. The **Add Relationship** dialog box displays, prompting you to add the selected item to a relationship field in the target item.

To save the selected relationship field as the default for the next time you add a related item using drag-and-drop, enable remember this field until the target view closes. The selected relationship field is saved as the default until the Items view closes.

To save the added relationship, click **OK**.

- You can also work with related items in the **Relationships** view.
- If you are viewing Relationships for a historical item, the following actions are disabled: add relationship flag, remove relationship flag, add related item, remove related item, move related item up, and move related item down.

## Related Item Tasks

You can perform the following tasks with related items:

Task	Procedure
Add a related item	<p>In the GUI, do one of the following:</p> <ul style="list-style-type: none"><li>■ When editing an item, click  next to the relationship table field.</li><li>■ From the Items view, select the item and select <b>Add Related Item</b>.</li></ul> <p>In the Web interface, select the item in the Items view, and select <b>Add Related Item</b>.</p> <p>Notes:</p> <ul style="list-style-type: none"><li>■ If your administrator has set up a default browse query for the relationship field and you have permission to view it, that query will be used when browsing for item IDs. Otherwise, the last used query is used. You can select a different query if required.</li><li>■ When you add a related item to the default <b>Tests</b> field, a dialog box displays, allowing you to select a test document, or select individual test cases from a test document.</li><li>■ In the GUI, you can also add related items from the Document view, Relationship view, and the Item Details view.</li></ul> <p>If the relationship field is in comma-separated values format (GUI only), type the related item ID in the field. If the relationship field allows multiple related items, separate the items with a comma.</p>
Create a new item as a related item	<p>When editing an item, click  next to the relationship field.</p> <p>By default, the <b>Create Relationship</b> option is enabled. If you do not want to relate the new item to the current item, clear this option.</p> <p>By default, the <b>Copy Common Fields</b> option is enabled. If you do not want to include the contents of common fields in the new related item, clear this option.</p> <p>Note: MKS Integrity determines common fields based on the same criteria as copying an item.</p> <p>In the <b>Add Relationship to Field</b> list, select the relationship field you want to use to relate the new item to the current item.</p> <p>If necessary, select a different item type to create using the data filter.</p> <p>You can also create a new item as a related item through the Items view.</p> <ul style="list-style-type: none"><li>■ In the GUI, select the item and select <b>New Related Item</b>.</li><li>■ In the Web interface, select the item in the Items view, and select <b>Create Related Item</b>.</li></ul> <p>Note: You can select multiple items in the Items view and create a related item that will be related to all of them. When you create a related item for multiple items, the <b>Create Relationship</b> and <b>Copy Common Fields</b> options are read-only. If you select items of different types, only relationship fields that are common to all types display in the <b>Add Relationship to Field</b> list.</p>
Add a relationship flag to a related item	<p>When editing an item:</p> <ul style="list-style-type: none"><li>■ In the GUI, select the related item and click  beside the field. The <b>Modify Relationship Flags</b> dialog box displays. Click <b>Add</b> and use the data filter to select the flag you want to add.</li><li>■ In the Web interface, select the related item and click  beside the field. The <b>Modify Flags</b> dialog box displays. Select <b>Add the flags below to the selected items</b>, then select the flags to add and click .</li></ul> <p>If the relationship field is in comma-separated values format (GUI only), type the relationship flag character directly in the relationship field after the appropriate item ID.</p>
Remove related item	<p>When editing an item, select the related item in the list, and click .</p> <p>If the relationship field is in comma-separated values format (GUI only), highlight the related item ID in the relationship field, and press <b>DELETE</b>.</p> <p>Note: Removing a related item does not delete the item; only the relationship within the database is removed.</p>

Task	Procedure
Remove relationship flag	<p>When editing an item:</p> <ul style="list-style-type: none"> <li>■ In the GUI, select the related item and click  beside the field. Click <b>Remove</b> and use the data filter to select the flag you want to remove.</li> <li>■ In the Web interface, select the related item and click  beside the field. Select <b>Remove the flags below from the selected items</b>, select the flags to remove, and click .</li> </ul> <p>If the relationship field is in comma-separated values display format (GUI only), highlight the flag character, and press <b>DELETE</b>.</p>
Copy and replace a shared test step	<p>You cannot edit a shared test step, but you can copy and replace it in the <b>Shared Test Steps</b> field. This enables you to use the contents of an existing test step but make modifications to it based on your test case.</p> <p>To copy and replace a shared test step when editing an item, select the shared test step and click  beside the field.</p> <p>After you save your changes, the new test step item replaces the copied test step item in the <b>Shared Test Steps</b> field.</p>

## Working With Attachment Fields

Your administrator determines the types of items that can have attachments. By default, attachments display in the **Attachments** panel, but may also display on other panels depending on how your administrator has defined the item type.

### Prerequisites

Depending on how your administrator defined the attachments field you may not be able to edit attachments. Conversely, your administrator may have set it up so multiple users can edit items and their related artifacts. To see your changes and the changes of others, you may need to refresh the window.

### Key Considerations

- By default, MKS Integrity allows attachments to be a maximum file size of 4 MB. Your administrator may define a higher or lower limit depending on the requirements of your system. Large attachments may affect performance.
- A summary is editable for an attachment only when the attachment is added to an item initially. You cannot edit a summary after you complete the operation.
- Attachments can be added or edited as rich content depending on how your administrator set it up. For more information, see “Working With Rich Content Fields” on page 24.
- If you are using the MKS Integrity GUI or Web interface, you must manually remove and re-add attachments to update the attachment version. When you manually remove and re-add an attachment in a single operation, MKS Integrity records that operation as **Attachments Updated** in the item history. The existing attachment is always replaced whether or not it has changed.

---

**IMPORTANT** The comparison of attachments is based on a message dialog algorithm (MD5) which relies on a checksum to calculate and compare the size of the attachment files. This comparison method is not guaranteed to work in all cases and it is possible to have collisions between two different attachments that are identical in size. While such collisions are rare, to avoid any possibility of collisions when using the `im editissue` command, you can manually update the attachments using the `addAttachment` and `removeAttachment` command options.

- You can also add an attachment using drag-and-drop. From your desktop or from Windows Explorer, drag the file you want to attach to an item in the Items view or Item Detail view.

- 
- Attachment file names containing an ampersand (&) are stored as %26 by MKS Integrity, even if the attachments are added through rich content fields or as a link.

### Attachment Tasks

You can perform the following tasks on attachments from the **Edit Item Details** dialog box:

Task	Procedure
Open an attachment	In the GUI, right-click the attachment, then select <b>Open</b> . In the Web interface, click the attachment name link.
Save an attachment	Right-click the attachment you want to save, then select <b>Save As</b> .
Add an attachment	Click  next to the attachment field. Enter the path and name of the file to attach, or browse to select a file. Click <b>Open</b> .
Replace an attachment (GUI only)	Right-click the attachment you want to replace, then select <b>Replace</b> . Enter the path and name of the file to attach, or browse to select a file. Click <b>Open</b> .
Delete/Remove an attachment	Select the attachment you want to remove and click  .

## Providing an Electronic Signature for an Item

If your administrator has set up *electronic signatures*, you may need to provide your user name and password when creating or making specific changes to an item. For example, you could be required to provide an electronic signature when you change an item's state to `Completed`. Signature information is recorded in the item's history.

To sign an item change, complete the **Signing User**, **Signing Password**, and **Signature Comment** fields, then click **OK**.

# Editing Items Inline (GUI Only)

You can easily edit item fields from the Items view in the GUI, without requiring the **Item > Edit** command or its subsequent dialog box. To do this you can:

- Enable inline editing
- Display the Edit Item Details view in the Items view.

---

**IMPORTANT** Other users can make changes to items that affect your Items view while you are editing items in it. To see their changes, select **View > Refresh**.

---

## ***Using Inline Editing in the GUI***

To enable inline editing, select **View > Options**, and select **Allow Table Editability**.

To edit an item field, click the field to select it (the data filter displays for applicable fields), then modify it according to its type.

An asterisk (\*) displays in the view title bar if there are unsaved changes to an item in the view.

---

**NOTE** You cannot inline edit an item if a relevance rule is set that prevents it. Contact your type administrator for more information.

---

Some fields include text formatting capabilities similar to those used in conventional word processor applications. Selecting such fields causes additional formatting menu items and toolbar buttons to become active and available.

---

**TIP** To follow a hyperlink in the field, press CTRL + click the hyperlink.

---

## ***Editing Items in the Edit Item Details Pane***

To enable editing in the Edit Item Details pane, select **View > Options**, and then select **Show item detail view** or **Show item edit view** on the **General** tab of the **Options** dialog box.

Editing items through the Edit Item Details pane instead of the Items pane enables you to edit all editable fields and correct errors in fields that do not display in the Items pane. If the pane is not displayed, it will display automatically whenever you need to resolve a mandatory field that isn't displayed in the Items view, as well as for problems with field relationship, event triggers, and editing conflicts.

A change made in the Items pane (using inline editing) is reflected in the Edit Item Details pane and a change made in the Edit Item Details pane is reflected in the Items pane if that field is displayed as a column. When the item is modified in either pane the view title bar shows the changed symbol (\*) and a  displays beside the list item and at the top of the detail pane until the changes are saved. Changes are saved when you move to another item in the list.

# Editing an Item

**CLI EQUIVALENT** im editissue

To add additional information to an item, you can edit the fields, and add attachments, relationships, and time entries. You can edit items depending upon the permissions assigned to you by your administrator.

For example, you would need to edit an item if someone on your project team assigned an item to you making you responsible for performing some action, or if you were responsible for reviewing all new submissions and assigning them to other team members.

## **Prerequisites**

To edit an item, ensure that you have the permissions to edit the item type and the relevant fields.

## **Editing Restrictions**

- Your administrator defines which item types and custom fields you are allowed to edit. If your administrator defines a field as a logging text field, you may only enter new text and not edit existing text.
- Depending on your workflow, you may not be able to edit an item that is in a terminal state.
- The item backed pick list relationship may be used to create a self-referencing editable field value attribute; however, you cannot set an item backed pick list field to the item it is currently located in.
- Field value relationships are not editable if no backing relationship is defined. Specifically, in the case where you are simultaneously modifying a relationship and field value attribute field, you must modify the relationship first, save the item, and then open it again to edit and save the field value attribute relationship.
- You cannot set a date field to null if the date has been previously set.
- You cannot select inactive values for user, group, project, or pick list fields. If a selected value becomes inactive and the field remains untouched during an item edit, the inactive value persists. If you modify the field with the inactive value, you must change it to an active value.
- If an item contains multi-valued user or group fields with inactive values, editing the inactive values and saving the item prompts you to leave the fields unchanged or clear the selections of all inactive values.
- Some options you see in the Item Details view are disabled for historical items. Historical versions of items are differentiated from current items by the As Of item header which contains the date and time you are viewing the item as of. For more information on historical items, see “Viewing a Historical Item” on page 42.

## **Key Considerations**

- If rich content field information cannot be displayed properly in a table, it displays as plain text.
- If an item contains a complex computed expression that needs to be evaluated on the MKS Integrity Server, editing the item and clicking **Save** does not re-calculate and update the new value in the item.

- 
- After viewing a tab in the Item Details, editing the item retains the focus on the last viewed tab, provided the tab exists in the view template and is not one of the following special tabs: **Workflow**, **History**, **Change Packages**, **Labels**, **Branches**, and **Test Results**. If the tab does not exist in the view template or is a special tab, editing an item shifts the focus to the first tab in the item.

#### To edit an item

In the Items view, select the item, and then select **Item > Item (GUI)** or **Edit Item (Web)**.

As soon as you enter or edit item information, an asterisk (\*) displays in the title bar until the changes are saved. To commit the item information to the database and resume entering additional information, click **Save**. To commit the item information to the database and close the item, click **Save & Close**.

## Configuring the Edit Item View

You can configure the look and behavior of the Edit Item view in the GUI by selecting **View > Settings**. The **Settings** dialog box displays.

Option	Description
<b>Buttons</b>	Show the <b>Save &amp; Close</b> , <b>Save</b> , <b>Cancel</b> , and <b>Help</b> buttons at the bottom of the view.
<b>Field constraints options</b>	Clear values that have been made invalid by a field relationship change. If a change to a source field value in a field relationship makes the value in the related target field invalid, the invalid value is cleared automatically. If this option is not checked, you are prompted with a message and can choose whether to clear the invalid value.

---

# Batch Editing Items

Batch editing allows you to simultaneously edit the fields that exist in multiple selected items. For example, you could change the assigned group for multiple items from Development to QA.

## **Prerequisites**

Before you batch edit items, ensure that you have the following:

- Permissions to edit the item type and relevant fields.
- If you are batch editing the **Assigned User** or **Assigned Group** fields, the list of users and groups you can select from is restricted to the allowed users and groups that are common to the listed projects.
- If you are batch editing the **Project** field, make sure the new project is one that the **Assigned User** has permission for; otherwise, he or she cannot edit the item.

## **Key Considerations**

- In the GUI, you can add related items and attachments to multiple items.
- Batch editing replaces the original content of a field with new content unless your administrator enables logging for that field in which case the new content appends the original content.
- Before you can batch edit items to change a state, you must first batch edit the mandatory fields that allow a state change. For example, if the **Root Cause** is a mandatory field for changing the state from `Accept` to `Schedule`, you must batch edit this field before batch editing the state.
- If you are batch editing a relationship field, you can add new relationships and relationship flags, but you cannot edit existing ones or create related items.
- If a batch edit causes an item change that requires an electronic signature, the batch edit fails.
- When batch editing items, it is possible for two or more users to add information to the items, make changes that have no impact on one another's edits, or select values that conflict with one another. For more information on resolving editing conflicts, see "Resolving Conflicts When Editing an Item" on page 37.
- You cannot select inactive values for user, group, project, or pick list fields. If a selected value becomes inactive and the field remains untouched during an item edit, the inactive value persists. If you modify the field with the inactive value, you must change it to an active value.

## **To batch edit multiple items**

- 1 In the Items view, select the items you want to edit, and select **Item > Edit (GUI)** or **Edit Item (Web)**.
- 2 Use the data filter to select the fields you want to edit.
- 3 Specify a new value for each field by using the data filter.
- 4 To accept the batch edits, click **OK**.

# Resolving Conflicts When Editing an Item

When multiple users make changes to an item at the same time, the first user's changes are saved. After the second user makes changes to the item and attempts to save them, MKS Integrity alerts the second user that another user also made changes to the item. The second user may not save the changes to the item until the editing conflicts are resolved.

## **Key Considerations**

- When resolving conflicts, normal editing constraints apply, for example, editability, relevance rules, and permissions.
- If another user edits the item containing the conflict you are resolving, the Editing Conflict dialog box may display after you resolve the current conflict.
- If the first user's edits cause a field's relevance or editability to change, when the Edit Item view is refreshed the new status of the field changes. If the second user was editing that field, they may not be able to resume editing after the view is refreshed. The value of the field always reverts to the committed value from the first edit. If the field is still relevant, ! displays. If it is no longer relevant, it does not display in the refreshed view.
- If the first user's edits cause the item to no longer be editable by the second user, then an error message displays and the second user must exit the edit. The second user does not have the option of returning to the Edit Item view.
- If there is a conflict with a logging text field, the two values are appended together and ! displays next to the field.
- If two users add an attachment with the same name, the refreshed item contains the attachment added by the first user. If the second user wants to re-add their attachment, a different name must be specified.
- Field value relationships are not editable when no backing relationship is specified. Specifically, if you are modifying a relationship field and a field value attribute field during the same editing session, you must modify the relationship field first, save the item, and then re-open to edit and save the field value relationship.
- Although the Relationships view allows you to edit multiple items from a single window, each item is sent to the server as a separate edit command. Each item edit command is handled like any other edit from the GUI. If a conflict occurs, the Editing Conflict dialog box displays and allows you to resolve the conflict.

## **To resolve conflicts when editing an item**

When editing an item and another user changes the same item before you save your changes, the **Editing Conflict** dialog box displays. The specific changes made by the other user are listed.

- 1 Review the changes made by the other user, then do one of the following:

- To automatically combine your changes with the other user's changes, click **Auto Resolve**.
- To discard your changes, click **Cancel**.
- To resolve the editing conflict, click **Resolve**. The Edit Item view displays the changes made by the other user. Each tab containing editing conflicts is annotated with an icon and each field containing editing conflicts is annotated with an icon. For more information on the different icons, see "Editing Conflict Icons" on page 38.

---

**TIP** Placing the cursor over an icon next to a field displays a detailed tooltip about the changes made to the field. Depending on the length of the changes, the tooltip may be truncated.

---

**2** Do one of the following:

- To commit the changes to the database and resume resolving additional conflicts, click **Save**.

---

**NOTE**

- Each **Save** operation is added to the item history.
  - When you commit item information to the database, fields in the item dynamically update based on relevance and editability rules. This means that what you can see and do in the item may change.
- 

- To commit the item information to the database and close the view, click **OK**. The revised item displays in the Item Detail view. For more information, see “Key Considerations” on page 37.

If another user edited the item while you were resolving conflicts, the **Editing Conflict** dialog box displays. Repeat steps 1 - 2.

### ***Editing Conflict Icons***

The legend at the top of the Edit Item view indicates what the following editing conflict icons mean:

Icon	Description	Resolution
	A value you edited was overwritten by another user.	To display options for changing the value, click the icon. <ul style="list-style-type: none"><li>■ If you disagree with the value, change it by selecting the desired value.</li><li>■ To approve the change, click <b>OK</b>.</li></ul> The icon changes to , indicating you have addressed the conflict.
	A value was changed by another user.	To view the change, click the icon. To approve the change, click <b>OK</b> . The icon changes to . Clicking the icon displays detailed information about the changes made by the other user.
	A value was changed by you.	No action required.

# Working With Item Change Packages

As part of MKS Integrity's software configuration management capabilities, change packages group changes to MKS Integrity members that were made as a result of the item. For example, a defect's change package might include the files that were changed in order to correct the problem. While viewing or editing an item, you can create, view, or close a change package for that item.

By default, MKS Integrity includes two change package types: MKS Integrity and Implementer. A change package type consists of change package attributes and change package entry attributes.

If you are using change packages in your environment and need more detailed information on using them with software configuration management projects, contact your administrator.

## Prerequisites

Your administrator defines the following:

- What item types can have change packages created for them.
- Who is allowed to create change packages for a specific item type.
- What item states allow open change packages.

For further assistance, contact your administrator.

## Change Package Tasks

The following table describes change package tasks for an item:

Task	Procedure
Create a change package for an item	In the GUI, click <b>Change Package &gt; Create</b> . In the Web interface, click  and select <b>Create Change Package</b> .
View an item's change package details	On the <b>Change Packages</b> tab of the <b>Item Details</b> view: <ul style="list-style-type: none"><li>■ In the GUI, select a change package and select <b>Change Package &gt; View Change Package Details</b>.</li><li>■ In the Web interface, select a change package and click the change package ID link.</li></ul> <p><b>Tip:</b> When viewing a change package you can edit the change package information if required, for example, to provide a more complete summary or description.</p>
Close an item's change package (GUI only)	Select a change package on the <b>Change Packages</b> tab and select <b>Change Package &gt; Close</b> . Closing a change package allows the item to move forward in the workflow.

# Working With Item Labels

Labels allow you to associate a name with a point in an item's history; for example, a prior release, project milestone, or document baseline. Labeling is disabled by default and is available primarily for the use with the ALM template.

## Prerequisites

Your administrator determines the types of items that can have labels and who can add labels to items.

For further assistance, contact your administrator.

## Label Tasks

The following table contains tasks you can perform for labels:

Task	Procedure
Add a label	<p>On the <b>Labels</b> tab in the <b>Item Details</b> view:</p> <ul style="list-style-type: none"><li>■ In the GUI, right-click in the labels field and select <b>Add Item Label</b>.</li><li>■ In the Web interface, click .</li></ul> <p>In the <b>Label</b> field, type the name of the label for the item. For example, <code>Release ABC</code>. In the <b>Comment</b> field, type a comment to associate with the label. Select the applicable <b>As Of</b> option from the list. The default date is the current date and time (Now). Click <b>OK</b>.</p> <p>The new label displays in the labels field. For more information about viewing historical item details, see "Viewing a Historical Item" on page 42.</p> <p>In the GUI, you can also add a label in the <b>Items</b> view by selecting an item and selecting <b>Add Item Label</b>.</p> <p><b>Tip:</b> In addition to adding the labels described below, you can also add item labels as of a modification date in the <b>History</b> tab by right-clicking a modification entry (GUI) or clicking the <b>Add Label</b> link beside the modification entry (Web).</p>
Move a label	<p>On the <b>Labels</b> tab in the <b>Item Details</b> view:</p> <ul style="list-style-type: none"><li>■ In the GUI, right-click in the labels field and select <b>Move Item Label</b>, and then select the label you want to move from the list.</li><li>■ In the Web interface, click .</li></ul> <p>Select the label you want to move. Select the <b>To Date</b> you want to move the label to. Click <b>OK</b>. The default date is the current date and time (Now). The <b>From Date</b> indicates the date associated with the label you are moving. The label details in the labels tab indicates the new date associated with the <b>As Of</b> option you selected.</p> <p>In the GUI, you can also move a label in the <b>Items</b> view by selecting an item and selecting <b>Move Item Label</b>.</p>
Delete a label	<p>On the <b>Labels</b> tab in the <b>Item Details</b> view:</p> <ul style="list-style-type: none"><li>■ In the GUI, right-click the label you want to delete and select <b>Delete Item Label</b>.</li><li>■ In the Web interface, select the label you want to delete and click .</li></ul> <p>In the GUI, you can also delete a label in the <b>Items</b> view by selecting an item, selecting <b>Delete Item Label</b>, and then selecting the item label you want to delete.</p>

---

# Copying an Item

---

**CLI EQUIVALENT** `im copyissue`

---

You can copy the common fields of an item and save them as a new item with a different name. Copying an item is useful when you want to spawn a new item from an existing item, retaining the existing information in the common fields. For example, if you want to spawn a Docs item from a Change Order item, copying the Change Order item retains the common fields and uses them in the new Docs item.

MKS Integrity determines common fields based on the following:

- Any fields defined as copy fields for the item type by your administrator that are visible to you in both the item type being copied and the new item type being created.
- If no copy fields have been defined by your administrator, all fields visible to you on both item types are copied, except for relationship and attachment fields (attachments referenced in rich content fields can be copied).

## **Prerequisites**

Before you copy an item, ensure that you have the permissions to create the new item type and fill out the relevant fields.

For further assistance, contact your administrator.

## **To copy an item**

- 1 From the Items view, select an item and then select **Item > Copy** (GUI) or **Copy Item** (Web).
- 2 If necessary, select a different item type.
- 3 Click **OK**.

---

**NOTE** You cannot save the copied item if it contains inactive values for user, group, project, or pick list fields. Change the values to active values before clicking **OK**.

---

- 4 Make changes to the new item and save it, as you would when creating a new item.

# Viewing a Historical Item

Historical items are versions of the current item associated with a date in the past. You can view historical items by label, branch, date, (user) edit. Historical items are denoted in the Item Details view with a title and header that indicates the date and time you are viewing the item **As Of**.

Task	Interface	Procedure
Viewing a historical item <b>Tip:</b> To learn more about As Of options for historical items, see "Historical Item Viewing Options" on page 42.	GUI/Web	<ul style="list-style-type: none"><li>■ Select a date hyperlink for an edit, label, or branch operation in the <b>History</b> tab of an Item Details view to view the item as of that date.</li></ul>
	GUI	<ul style="list-style-type: none"><li>■ Select the item you want to view from the Items view and select <b>Item &gt; Historical &gt; View Item Details As Of</b>.</li><li>■ You can view an item as of a label, branch, date, or edit. The Item Detail view displays the historical item details as indicated by the As Of date and time in the header.</li></ul>
	Web	<ul style="list-style-type: none"><li>■ Select the item you want to view from the Items view, click  and select <b>Historical &gt; View Item As Of</b>.</li><li>■ You can view an item as of a label, branch, date, or edit. The Item Detail view displays the historical item details as indicated by the As Of date and time in the header.</li></ul>

## Key Considerations

- Actions performed on historical items are performed against the specified or historical date.
- You cannot edit a historical item.
- All branches, labels, and item histories display in a historical Item view.
- You cannot create a change package for a historical item.
- Historical items are printed as of the historical date indicated in the header of the Item Detail view. Change packages and time entries are not printed.
- The **Workflow**, **Change Packages**, and **Time Entries** tabs do not display on the item details view of a historical item.
- Relationships display as of the current or historical date indicated on the item.
- Attachments display as of the historical date.
- Running a report on a historical item runs the report as of the historical date.
- To display **As Of** item edits in the **View Historic Item** dialog box for a Requirement Document in the document model, the item must contain edits to fields in the significant edit fields list for the content type. For more information, see your administrator.

Shared item types also have a significant edit field list; however, MKS Integrity does not make assumptions about significant edits on the node, based on those of the shared item. For more information on the node and shared item, contact your administrator about working with related items in a document structure.

- When viewing historical items containing computed fields, if the computed field was set up by your administrator to calculate and store to history on a daily, weekly, or monthly basis, the most recently stored values display.

## Historical Item Viewing Options

The following table lists the options for viewing items **As Of**:

---

To view an item As Of...	Do this from the As Of list...
Label	Select <b>Label</b> , select the label you want to view the item as of, and click <b>OK</b> .
Branch	Select <b>Branch</b> , select the branch you want to view the item as of, and click <b>OK</b> .
Date	Select <b>Date</b> , select a date to view the item as of, and click <b>OK</b> .
Edit	Select <b>Edit</b> , select the edit you want to view the item details as of, and click <b>OK</b> .
Now	Select <b>Now</b> and click <b>OK</b> . <b>Note:</b> Viewing an item as of <b>Now</b> is the same as viewing an item with no <b>As Of</b> date.

# Relationships View: Viewing Item Relationships

The Relationships view displays all the relationships for one or more items in a tree hierarchy. You can configure the Relationships view to display specific fields and relationship levels.

## To display the Relationships view in the GUI

In the GUI, select one or more items in the Items view or Item Details view, then select **Item > View Relationships**.

## To display the Relationships view in the Web interface

Do one of the following:

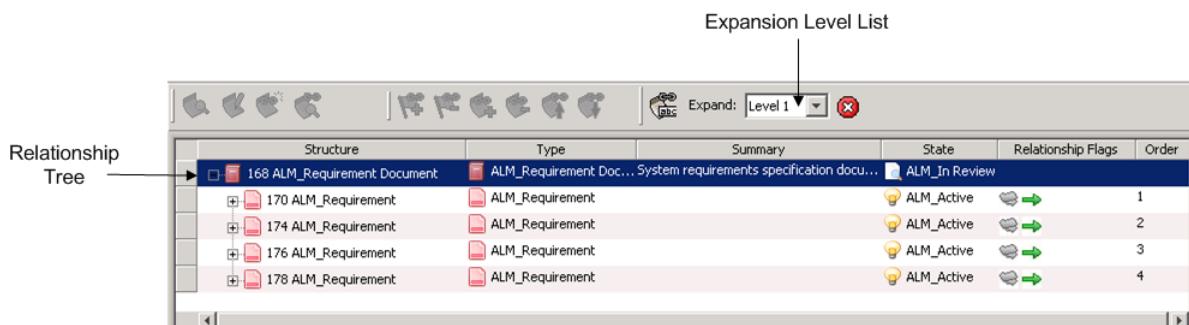
- Select one or more items in the Items view or Item Details view, and select **View Related Items**.
- At the top of the Items view, click  and then one of the following options:
  - **Don't show relationships** hides the Relationships view (if visible). By default, the Relationships view is hidden.
  - **Show in Side bar** displays the Relationships view in a frame to the right of the Items pane and/or Item Details pane.
  - **Show in Footer** displays the Relationships view in a frame below the Items pane and/or Item Details pane.
  - **Show in Detail view** displays the Relationships view in a frame below the Item Details pane (only if visible).

**Show selection in detail pane** dynamically updates the Item Details pane and/or Relationships view based on your selection in the Items pane or Relationships view.

## Relationships View: User Interface Components

When the Relationships view first displays, the Relationships tree pane shows the item(s) you selected.

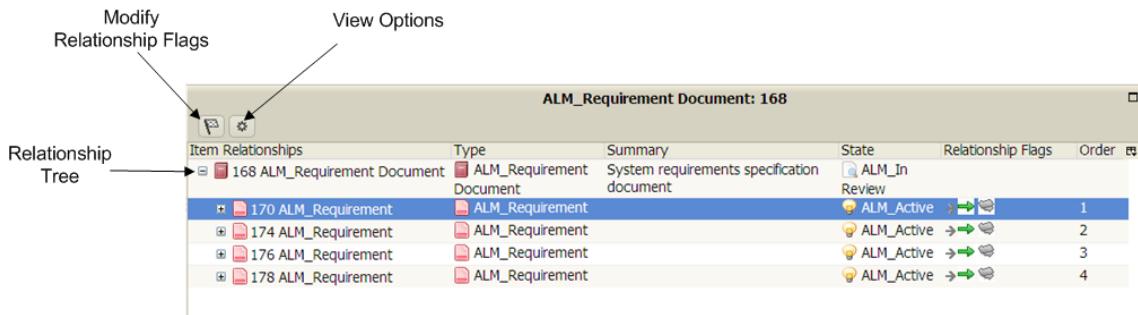
The following is an example of an item and its relationships in the Relationships view (GUI):



The screenshot shows the Relationships view in the GUI. On the left is a 'Relationship Tree' pane containing a tree structure with several items. On the right is a 'Grid' pane displaying detailed information for each item. An arrow points from the text 'Relationship Tree' to the tree icon in the tree pane. Another arrow points from the text 'Expansion Level List' to the 'Expand: Level 1' dropdown menu in the grid pane's header.

Structure	Type	Summary	State	Relationship Flags	Order
168 ALM_Requirement Document	ALM_Requirement Doc...	System requirements specification docu...	ALM_In Review		
170 ALM_Requirement	ALM_Requirement		ALM_Active	→	1
174 ALM_Requirement	ALM_Requirement		ALM_Active	→	2
176 ALM_Requirement	ALM_Requirement		ALM_Active	→	3
178 ALM_Requirement	ALM_Requirement		ALM_Active	→	4

The following is an example of an item and its relationships in the Relationships view (Web):



The Relationships view includes the following user interface components that allow you to understand the item's relationships:

Component	Description
Relationship Tree	You can expand the tree to show the relationship fields for the root items by clicking the plus icon. You can expand each relationship field to see the related items in that field, expand the related items to see their relationship fields, and so on.
Expansion Level List (GUI only)	The Expansion Level list allows you to expand or retract all relationships in the Relationship tree pane to a specific level for a given item. In default and migrated ViewSets, floating Relationship views display the Expansion Level list in the <b>Expansion Panel</b> toolbar by default. If you are creating a ViewSet, you must add the <b>Expansion Levels</b> action to a toolbar. If you attempt to expand the tree, and the number of relationships is taking too long to display, click  to stop the expansion. To collapse a branch of the tree, click .
	If an item contains a relationship loop,  displays beside the item. A relationship loop occurs when an item has both a forward and a backward relationship with another item within the same relationship hierarchy.
Tip:	<ul style="list-style-type: none"> <li>■ In the GUI, you can also expand and collapse the tree by selecting a branch and using the right or left arrow keys.</li> <li>■ In the GUI, you can drag an item from the Items view to an item's <b>Forward Relationships</b> or <b>Backward Relationships</b> folder in the <b>Relationships</b> view to create a new relationship.</li> </ul>
Modify Relationship Flags (Web only)	Adds or removes link flags for the selected item. For more information, see "Working With Items in the Relationships View" on page 45.
View Options (Web only)	Configures the look of the Relationships view, including what view components display, how the item information displays, and what relationships display in the tree. For more information, see "Configuring the Relationships View" on page 47.

## Working With Items in the Relationships View

When the Relationships view is expanded to show related items, you can perform tasks on an item by selecting it and then performing a procedure described in the following table:

**Note** If you are viewing Relationships for a historical item, relationships display for the item based on an **As Of** date indicated in the Item Detail view header. For more information about historical items, see "Viewing a Historical Item" on page 42.

Task	Procedure
Open another <b>Items</b> view (GUI only)	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; View Items</b>.</li> </ul>
View details for item	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; View Item Details</b> (GUI).</li> <li>■ Select <b>View Item</b> (Web).</li> </ul>
Open a selected item's <b>Relationships</b> view	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; View Relationships</b> (GUI).</li> <li>■ Select <b>View Related Items</b> (Web).</li> </ul>
Create new item (GUI only)	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Create</b>.</li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>■ Select <b>Item &gt; New &gt; &lt;item type&gt;</b>.</li> </ul>
Copy selected item	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Copy</b> (GUI).</li> <li>■ Select <b>Copy Item</b> (Web).</li> </ul>
Create new item and add it as related item	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Create Related</b> (GUI).</li> <li>■ Select <b>Create Related Item</b> (Web).</li> </ul>
Delete selected item (GUI only)	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Delete</b> (GUI).</li> </ul>
Edit selected item	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Edit</b> (GUI).</li> <li>■ Select <b>Edit Item</b> (Web).</li> </ul>
Print selected item (GUI only)	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Print</b>.</li> </ul>
Run report for selected item	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Run Report</b> (GUI).</li> <li>■ Click  and select <b>Run Item Report</b> (Web).</li> </ul>
Run a historical report (Web only)	<ul style="list-style-type: none"> <li>■ Click  and select <b>Historical &gt; Run Report As Of</b>.</li> </ul>
Run an item differences report (Web only)	<ul style="list-style-type: none"> <li>■ Click  and select <b>Run item differences report</b>.</li> </ul>
Add existing related item	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Relationship &gt; Add Related Item</b> (GUI)</li> <li>■ Open the Items view (<b>Item &gt; View Items</b>), select an item, and drag it to a relationship field in the tree (GUI)</li> <li>■ Select <b>Add Related Item</b> (Web).</li> </ul>
Remove related item	<ul style="list-style-type: none"> <li>■ Select <b>Item &gt; Relationship &gt; Remove Related Item</b> (GUI).</li> <li>■ Select <b>Remove Related Items</b> (Web).</li> </ul>
Add or remove link flags for selected item	<ul style="list-style-type: none"> <li>■ In the GUI, select <b>Item &gt; Relationship &gt; Add Relationship Flag</b> or <b>Item &gt; Relationship &gt; Remove Relationship Flag</b>.</li> <li>■ In the Web interface, click  . Select <b>Add the flags below to the selected items</b>, then select the flags to add and click  , or select <b>Remove the flags below from the selected items</b>, then select the flags to remove and click  .</li> </ul> <p><b>Note:</b> You can add or remove link flags for multiple related items if they are all related through the same relationship field.</p>
Reorder selected related item in relationship field	<ul style="list-style-type: none"> <li>■ In the GUI, select <b>Item &gt; Relationship &gt; Move Related Item Up</b> or <b>Move Related Item Down</b>.</li> <li>■ In the Web interface, select <b>Move related item up</b> or <b>Move related item down</b>.</li> </ul>
Refresh items (GUI only)	<ul style="list-style-type: none"> <li>■ Select <b>View &gt; Refresh</b> (GUI).</li> <li>■ Select <b>Refresh View</b> (Web).</li> </ul>

## Drag-and-Drop Operations in the Relationships View (GUI)

In the Relationships view (GUI only), you can perform the following relationship operations using the drag-and-drop method:

Task	Procedure
Add a new relationship to an item while retaining an existing relationship	<p>Holding the ALT key, click the item you want to add as a new related item, and drag it to the target item. An arrow icon and <b>As Relationship</b> tooltip display.</p> <p>Release the ALT key and mouse button. You are prompted to move the selected item to a relationship field in the target item.</p> <p>To save the selected relationship field as the default for the next time you add a related item using drag-and-drop, enable <b>remember this field until the target view closes</b>. The selected relationship field is saved as the default until the Relationships view closes.</p> <p>To save the added relationship, click <b>OK</b>.</p> <p><b>Note:</b> If the relationship is single-valued, it must be visible on the item type for drag-and-drop to work.</p>
Move an existing relationship to a new item while deleting the old relationship	<p>Click the item you want to move as a related item, and drag it to the target item you want to relate it to. An arrow icon and <b>As Relationship</b> tooltip display.</p> <p>Release the mouse button. You are prompted to move the selected item to a relationship field in the target item.</p> <p>To save the selected relationship field as the default for the next time you move a related item using drag-and-drop, enable <b>remember this field until the target view closes</b>. The selected relationship field is saved as the default until the Relationships view closes.</p> <p>To save the moved relationship, click <b>OK</b>.</p> <p><b>Note:</b> If the relationship is single-valued, it must be visible on the item type for drag-and-drop to work.</p>

## Configuring the Relationships View

You can configure the look of the Relationships view, including what view components display, how the item information displays, and what relationships display in the tree.

### To configure the view components and item information in the GUI

- 1 Select **View > Options**. The **Options** dialog box displays.
- 2 On the **General** tab, select from the following options:
  - **View layout**
    - **Show all forward relationship fields** displays all fields that contain forward relationships. For more information, see “Working With Items in the Relationships View” on page 45.
    - **Show all backward relationship fields** displays all fields that contain backward relationships.
    - **Show field grouping rows** displays the name of the relationship field as a grouping row in the tree.
    - **Substitute parameters** displays parameter values for parameter references in text fields.
  - **Viewing or editing panel**
    - **Off** displays the **Items** pane only.
    - **Show item edit view** enables you to edit item details for the item selected in the **Tree** pane. You cannot select this option in combination with the **Show item detail view**.
    - **Show item detail view** enables you to view item details for the item selected in the **Tree** pane. You cannot select this option in combination with the **Show item edit view**.
- 3 By default, long field values in tooltips may be truncated. You can display the full value by enabling **Show full field value in tooltip**.

- 
- 4 On the **Titles** tab, click **Fields** to select the fields that you want to display for each item node. The field names are added with "{" and "}".
  - 5 On the **Messages** tab, **Show confirmation message before saving a modified item** displays a message when you save a modified item asking you to confirm saving the change.

#### To configure the relationships fields that display in the tree in the GUI

- 1 Select **View > Filter Fields**.
- 2 In the **Filter Displayed Fields** dialog box, select the relationship fields you want to display in the tree.

#### To configure the relationships view in the Web interface

- 1 Click . The **Configure View** dialog box displays.
- 2 Select one of the following:
  - **Consider only these relationship fields to produce the tree:**  
Click  and select the relationship fields that you want to display.  
To remove a relationship field from the tree display, select it and click .  
To change the display order of the relationship fields in the tree, click  and .
  - **Consider only relationships that are:**  
Specify whether you want to display forward relationship fields, backward relationship fields, or both.
- 3 Select **Show relationship fields in the tree** to display the name of the relationship field above the related items in the relationship tree.
- 4 Click the **Item Relationships Format** tab.
- 5 Click  to select the fields that you want to display for each node. The field names are added with "{" and "}".
- 6 Click **OK**.

# Printing an Item

---

**CLI EQUIVALENT** im printissue

---

You can print the details of an item. In the Items view in the GUI, you can also print a list of items. You can also customize the printout of your items.

In the GUI, customize the printout of your items through your client options.

In the Web interface, customize the printout of your items by clicking **My Profile** on the right side of the title pane, then clicking on the **Print Options** tab in the **My Profile** dialog box.

---

**NOTE** When the view is maximized, click  in the title pane and select **My Profile**.

---

Task	Procedure
Print item in GUI	Select item you want to print, and select <b>Item &gt; Print</b> , and then select <b>Selected Item Details</b> as the printout style.
Print listing of items in GUI	Select items you want to print, select <b>View &gt; Print</b> , and then select <b>Query Results List</b> as the printout style.
Print item in Web	Note the following before you use this command: ■ You may be prompted for your credentials. ■ If you are using a proxy server, the print preview may not display. From the Items view, select item(s) you want to print, and select <b>Print Item</b> , then select <b>File &gt; Print</b> from the new browser window.

---

# Receiving E-mail Notifications About Items

E-mail notifications allow MKS Integrity users to be notified by e-mail whenever a new item is submitted or item information changes. This is useful for staying informed about assigned items, new items that require review and approval, or state changes on existing items. More importantly, notifications allow you to respond immediately to critical items and stay informed on project progress.

The e-mail notification displays information about the item, for example, the item type, ID, summary, who edited the item and when, a hyperlink to the item, and the modified fields.

Your administrator can define additional fields to display in e-mail notifications for specific item types.

You configure MKS Integrity to send you e-mail notification by creating *rules*.

## **Rule Format (GUI)**

The Web interface allows you to set simple e-mail notification rules, for example, receiving e-mail notification when new items are assigned to you. To create advanced e-mail notification rules, you must create them in the GUI.

A *rule* is a statement that sets a specified outcome when certain conditions are met. In the GUI, rules are composed of nodes and conditions. *Nodes* are the logical connectors that describe the relationship between two statements (or conditions). *Conditions* are a statement of the requirements that must be satisfied, and can involve either user or field values.

For example, you could create a simple rule containing one condition that sends you e-mail every time a new problem is submitted that is assigned to you. Similarly, you could create a complex rule containing two conditions that sends you e-mail every time a new problem is submitted that is assigned to you, and when existing problems become assigned to you. Rules can contain as many conditions as you want.

## **Operators**

An operator is a function used to create e-mail notification rules in the GUI. The meaning of the operator depends on whether the field you are using in the filter or rule is single-valued or multi-valued, for example, the = operator means "equals" for single-valued fields and "contains" for multi-valued fields.

## **Nodes (GUI)**

The logical **and** node indicates that all of the specified conditions must be true to meet the requirements of the rule.

The logical **or** node indicates that one or more of the specified conditions must be true to meet the requirements of the rule.

The specific placement of the logical node is important to determining how it affects the meaning of the rule.

## **Example**

The following example shows an e-mail notification rule that will notify the user (administrator) each time a new change request is created or whenever a defect is assigned to the user. With the **or** node, the notification occurs whenever *either* of the events occurs.



The use of [New Value] in a rule condition indicates a change in a field value. One part of the condition indicates the field value before the change, and the other part indicates the value after the change. For example, in the rule above, Assigned User <> Assigned User[New Value] indicates that the new value of the Assigned User field is not equal to what it was before the item was saved. In other words, the value of the Assigned User field was changed during the item edit.

### **Prerequisites**

- Depending on how MKS Integrity is configured, your administrator may be the only one who can create and edit e-mail notification rules for users and groups. If you want to use the notification feature, you must have the following permissions assigned to you by your administrator:
  - allow ViewMyNotification allows you to view e-mail notification settings, but not save changes made to them.
  - allow ModifyMyNotification allows you to create and edit notification rules.
- E-mail notification is subject to project, type, and field visibility rules. Only users that have visibility for a given project and type receive e-mail notification for items related to that project and type. In addition, e-mail notifications include only the fields they have permission to view.

### **Key Considerations**

- MKS Integrity project fields cannot be used in e-mail notification rules.
- E-mail notifications are evaluated on the MKS Integrity Server's time zone; however, symbolic dates in rules are evaluated on the MKS Integrity Client's time zone.

### **To set e-mail notification in the GUI**

- 1 Select **Item > Set Notifications**.
- 2 The e-mail address specified for you by your administrator is automatically entered in the **Email Address** field.
- 3 Select **And** or **Or**, depending on the type of logical connector you want to use between conditions.

---

**NOTE** If you are creating a rule with only one condition, you do not need to select **And** or **Or**.

---

**Swap** replaces the selected node with the opposite node. For example, swapping an **Or** node replaces it with an **And** node.

**Remove** deletes the selected node.

- 4 Under **Condition**, select how you want to compare the value of a field.
- 5 Select the operator from the list.
- 6 Specify fields and field values for the condition.

---

**NOTE** You cannot specify MKS Integrity project and attachment fields in conditions.

---

- a) To choose a value for a date field, click **Change**.

b) Do one of the following:

- Select a date from the calendar. If the date field is configured to display the time and you want to include it, select the **Show time** option (if not already enabled) and include a time from the calendar. The **Show time** option is enabled by default.
- Select **now** to specify the current date and time. This option displays only if the date field is configured to display the date and time.
- Select **today** to specify the current date and a time of 00:00:00 (midnight). This option can be specified for a date field or a date field configured to display the date and time.
- Select **none** to specify an empty value for the date field.

7 For each field, specify an **Existing Value** or a **New Value**.

8 When you are finished constructing the condition, click **Add**. The condition displays under the selected node.

### To set e-mail notification in the Web interface

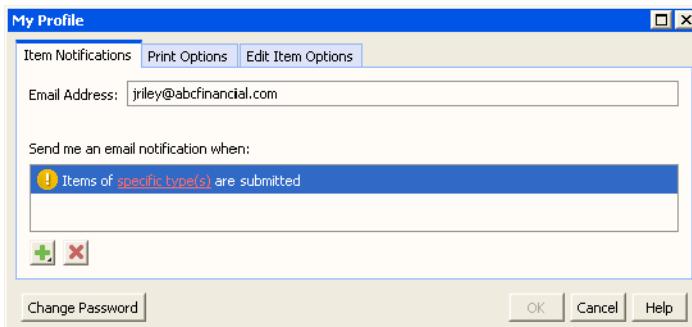
The Web interface allows you to set simple e-mail notification rules, for example, receiving e-mail notification when new submitted items are assigned to you.

**NOTE** To create advanced e-mail notification rules, you must create them in the GUI.

1 Click **My Profile** on the right side of the title pane. Your e-mail address displays on the **Item Notifications** tab.

**NOTE** If you create a notification rule in the GUI that is too complex to view or edit in the Web interface, you are asked whether you want to delete the existing rule and create a new one in the Web interface.

2 Click  and select a pre-defined notification rule. In the list,  displays next to the pre-defined rule.

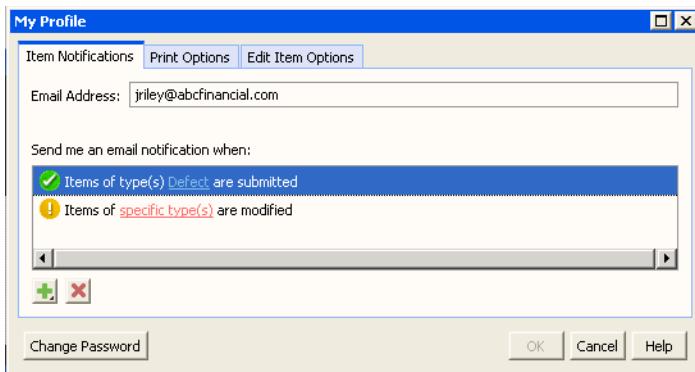


3 Click **specific type(s)** in the rule and use the data filter to select one or more item types to be notified about when the condition described in the rule is satisfied. For example, if you set the notification rule **Items of type(s) Defect are submitted** you would be notified when a new Defect item type is submitted.

If you select a rule that has an item state condition, click **specific state(s)** and use the data filter to select one or more states that the specified item type must be in for a notification to be sent. For example, if you set the notification rule **Items of type(s) Docs are assigned to state(s) Rejected** you

would be notified when a documentation item is rejected. The list of states you can select depends on the workflow implemented by your administrator.

If you add more than one rule at a time without defining any values for them, ! displays next to all rules. After defining the values for one of the rules, the icon changes to ✓.



- 4 Repeat the necessary steps for any additional notification rules. All rules are connected using the logical **or**, which indicates that one or more of the specified conditions must be true.

To remove a notification rule, select it and click X.

- 5 When all the notification rules are set, click **OK**.

# Recording Effort With Time Entries

The amount of development effort required to facilitate change during the development cycle can be recorded in individual items using *time entries*. This capability allows you to record time for work when it is required by your organization and provides data for planning future projects.

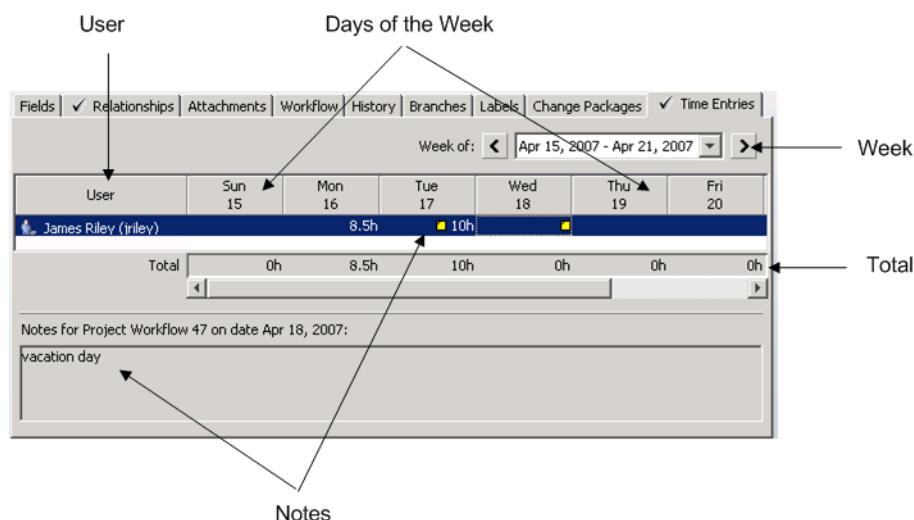
You can manage time entries from the following views:

- **Time Entries** tab in an Item Details view. This view manages time entries for a specific item, for example, the amount of time spent working on a Project or Change Order item during the week.
- **Time Entries** view. This view is the central location for managing all of your time entries. For example, you can view the time recorded for several different items during the week.

If you have the appropriate permission, you can manage time entries belonging to other users.

## Time Entries View: User Interface Components

The following is an example of time entries in the Time Entries tab:



The Time Entries tab and Time Entries view contain the following common interface components:

Component	Description
User	The <b>Time Entries</b> panel displays the currently logged in user under the <b>User</b> column. If the user has an e-mail address, the name displays as a hyperlink that when clicked starts the default e-mail program and creates an e-mail with the name of the item and a hyperlink to the item details.
Days of the Week	Date columns, for example, <b>Mon 1</b> . When creating or editing a time entry, press the TAB key to move to the date column to the right, and SHIFT + TAB to move to the date column to the left. Displayed time entries vary based on the time zone a user operates in, using the date format specified by your locale.
Week	List that displays the current week by default. To specify the previous or following week, click < or > beside <b>Week of</b> . To specify the week of a specific month or year, click the displayed date, and select a date from the calendar. As you move forward from week to week in the view, the previous week's rows containing time entries are carried forward and display in the current week. For example, if you add a time entry to Defect 131 on January 7, 2010, Defect 131 displays as an empty row in the week starting January 26, 2010.

---

Component	Description
Total	<b>Total</b> columns and rows that display the total time recorded for each day and week.
Notes	<b>Notes</b> field displays notes added to a time entry, indicated by  next to the duration. Selecting the time entry containing the note displays the contents of the note in the <b>Notes</b> field. You can also position your cursor over the yellow document icon to display the contents of the note as a tooltip. The <b>Total</b> row and column display the new time entry totals.

---

# Time Entries View: Managing Time Entries

Use the Time Entries view to manage all of your time entries or the time entries of other users.

## **Prerequisites**

- To edit or delete the time entries of another user, you must have the `TimeTrackingAdmin` permission assigned to you.
- To view time entries belonging to another user, the time entries must be recorded in a type or project that is visible to you.

In the Time Entries view, you can perform the following tasks:

Task	Interface	Procedure
View and Edit the details of an item associated with a time entry	GUI only	Select the time entry, then select <b>Item &gt; View Item Details</b> . or Select the time entry, then select <b>Item &gt; Edit</b> .
View time entries belonging to another user	GUI	Select <b>View &gt; Change User</b> , select a user, and then click <b>OK</b> .
	Web only	In the <b>Select User</b> field at the top of the view, select a new user, and then click <b>OK</b> .
Create time entries	GUI/Web	See “Creating a Time Entry” on page 57.
Edit or delete time entries	GUI/Web	See “Editing Time Entries” on page 59.

# Creating a Time Entry

**CLI EQUIVALENT** `im settimeentries`

Once an item has been created, you can record time spent working on the item by creating a *time entry*. A time entry records the duration of time spent working on the item in hours for a specific day. You can also include any relevant notes about the time spent working on the item, for example, designed regression test.

## Prerequisites

- An item does not have to be assigned to you in order to create a time entry; however, you must have visibility to the item.
- If the `TimeTrackingAdmin` permission is assigned to you, you can also create time entries for other users in the **Time Entries** view. This is useful if your organization has a designated role for recording employee work hours.
- Your administrator configures which types and states allow time entries.

## Example

Chad is assigned to a new Feature item for an ABC Financial client. For billing purposes, the project manager wants Chad to log the amount of time spent working on the item. After each day spent working on the item, Chad opens the Feature item for editing and logs the amount of time in hours.

## To create a time entry

Task	Interface	Procedure
Create a time entry when editing an item	GUI	<p>Open an item for editing and click the <b>Time Entries</b> tab. In the desired row, select the date cell you want to record time to, and type the amount of time spent working on the item in hours. All time entries are formatted to two decimal places.</p> <p>Optionally, you can include notes in the <b>Notes</b> field after you specify a duration.</p>
	Web	<p>Open an item for editing and click the <b>Time Entries</b> tab. In the desired row, click the date cell you want to record time to, and type the amount of time spent working on the item in hours. All time entries are formatted to two decimal places.</p> <p>Optionally, you can include notes in the <b>Notes</b> field after you specify a duration.</p>

Task	Interface	Procedure
Create a time entry from the Time Entries view	GUI	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>■ If you are creating a personal time entry, select <b>Time Entry &gt; Edit My Time Entries</b>.</li> <li>■ If you are creating a time entry for another user, select <b>Time Entry &gt; Edit Time Entries</b>.</li> </ul> <p>Select a user using the data filter, and then click <b>OK</b>.</p> <p>Under the <b>Item</b> column, click the <b>&lt;new&gt;</b> cell, then type the ID of the item that you want to add time entries to, or click the browse button to select one or more items.</p> <p>Select one or more items that you want to add time entries to, or run a query from the <b>Query</b> list to locate the specified items.</p> <p>Click <b>OK</b>. One or more items display under the <b>Item</b> column. In the desired row, select the date cell you want to record time to, for example, <b>Wed 3</b>. The view updates to display the item you specified and a new blank row to specify a new item.</p> <p>Type the amount of time spent working on the item in hours. All time entries are formatted to two decimal places.</p> <p>Optionally, you can include notes in the <b>Notes</b> field after you specify a duration.</p> <p>To commit the time entry information to the database and close the view, click <b>OK</b>. If you have the <b>My Time Entries</b> or <b>Time Entries</b> view open, the <b>Total</b> row and column display the new time entry totals. If you added a note to the time entry,  displays next to the duration indicating that it contains a note.</p>
	Web	<p>Click <b>Time Entries</b> on the right side of the title pane.</p> <p>If you are creating a time entry for another user, select a user using the data filter, and then click <b>OK</b>. Time entries (if any) for the selected user display in the <b>Time Entries</b> view.</p> <p>Under the <b>Item</b> column, click the <b>&lt;new&gt;</b> cell, then type the ID of the item that you want to add time entries to, or click the browse button to select one or more items.</p> <p>In the desired row, click the date cell you want to record time to, and type the amount of time spent working on the item in hours. All time entries are formatted to two decimal places.</p> <p>Optionally, you can include notes in the <b>Notes</b> field after you specify a duration.</p> <p>When you click <b>Save</b>,  displays next to the time entry indicating that it contains a note and the <b>Total</b> row and column display the new time entry totals.</p> <p>To commit the time entry information to the database and close the view, click <b>OK</b>.</p>

# Editing Time Entries

---

**CLI EQUIVALENT** im settimeentries

---

You can edit your existing personal time entries to correct durations and notes. During an edit, you can also delete time entries by clearing date cells and notes for the time period.

## Prerequisites

- To edit or delete another user's time entries, you must have the `TimeTrackingAdmin` permission assigned to you.
- An item does not have to be assigned to you in order to edit or delete existing time entries; however, you must have visibility to the item.
- In an **Edit <item>** view, you can only edit or delete personal time entries.

The following table describes how to edit and delete time entries:

Operation	Interface	Procedure
Edit or delete a personal time entry	GUI	<p>Do one of the following:</p> <ul style="list-style-type: none"><li>■ Open an item for editing, and click the <b>Time Entries</b> tab.</li><li>■ Select <b>Time Entry &gt; Edit My Time Entries</b>.</li></ul> <p>Select the time entry you want to edit or delete, then edit the information as required, or delete the time entry by clearing the date cells and notes.</p>
	Web	<p>Do one of the following:</p> <ul style="list-style-type: none"><li>■ Open an item for editing, and click the <b>Time Entries</b> tab.</li><li>■ Click <b>Time Entries</b> on the right side of the title pane.</li></ul> <p>Click the time entry you want to edit or delete, and edit the information as required, or delete the time entry by clearing the date cells and notes.</p>
Edit or delete another user's time entry	GUI	<p>Select <b>Time Entry &gt; Edit Time Entries</b>, select a user, and then click <b>OK</b>.</p> <p>Select the time entry you want to edit or delete, and edit the information as required, or delete the time entry by clearing the date cells and notes.</p>
	Web	<p>Click <b>Time Entries</b> on the right side of the title pane, and select a user in the <b>Select User</b> field at the top left of the view.</p> <p>Select the time entry you want to edit or delete, and edit the information as required, or delete the time entry by clearing the date cells and notes.</p>

---

# Viewing Time Entries

---

**CLI EQUIVALENT** `im timeentries`

---

All time entries for a specific user can be viewed in a single view. Any user can view another user's time entries if the time entries are recorded in a type or project that is visible to the user.

The following table describes how to view time entries:

Task	Interface	Procedure
To view your personal time entries	GUI	Select <b>Time Entry &gt; View My Time Entries</b> .
	Web	Click <b>Time Entries</b> on the right side of the title pane.
To view time entries for other users	GUI	Select <b>Time Entry &gt; View Time Entries</b> . In the <b>Select User</b> dialog, select the user you want to view time entries for.
	Web	Click <b>Time Entries</b> on the right side of the title pane. In the <b>Time Entries</b> view, use the <b>Select User</b> field to select the user you want to view time entries for.

## PART 2

# Displaying Items Meeting Specific Conditions

---

# What Is a Query?

A *query* is a request to select and list the items that meet specific selection conditions. You can create any number of queries, each with different conditions. This allows you to break items down into manageable groups according to your needs. For example, you may have one query to list items assigned to you that are currently in development, and another to list items assigned to you that are currently in review. Queries are also used as the basis for reports and charts, and can be included in dashboards.

You can define the following types of queries:

A *Quick Query* is an easily defined, disposable query that can only be used by you. A Quick Query is useful for creating temporary queries. By default, the Quick Query searches for all items assigned to you.

A *named query* is a query that can be given a name and description, and shared with groups. This allows you to create a query that applies to all members of a specific group, for example, a query that searches for all defects assigned to the development group.

A *sub-query* is a named query that is inserted as a filter into another query. More than one sub-query may be inserted into a query.

---

# Queries View: Managing Your Queries

---

## CLI EQUIVALENT `im queries`

---

You can view and manage all queries created by you or shared to you from the **Manage Queries** view (GUI) or **Queries View** (Web).

### To display the Queries view

Interface	Procedure
GUI	Select <b>Query &gt; Manage Queries</b> .
Web	Under <b>Quick Views</b> , select <b>Queries</b> .

From the Queries view, you can perform the following tasks:

Task	For more information, see...
Create a query.	“Creating a Query” on page 68.
View a query definition or edit a query.	“Editing a Query” on page 77.
Copy the details of an existing query into a new query.	“Copying a Query” on page 79.
Delete a query that is no longer used.	“Deleting a Query” on page 80.
Run a query.	“Running a Query” on page 78.

---

# Query/Query Definition View

---

**CLI COMMAND:** im viewquery

---

This view displays the definition of a specific query.

To display the **Query Definition** view in the GUI, select a query in the **Manage Queries** view, or run a query in the **Items** view and select **Query > View Definition**.

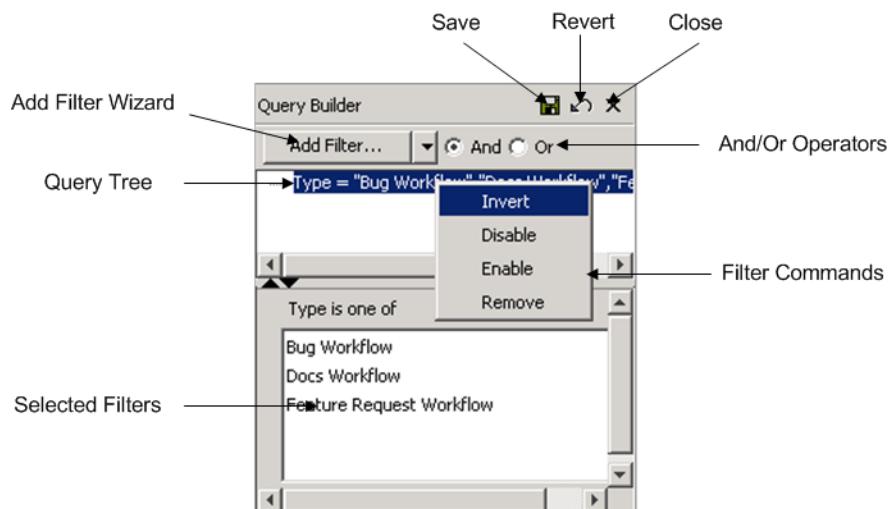
To display the **Query View** in the Web interface, select a query in the **Queries View**, click , and select **View Query Definition**.

To view specific information about the query, click a tab. You cannot edit the query definition while viewing it.

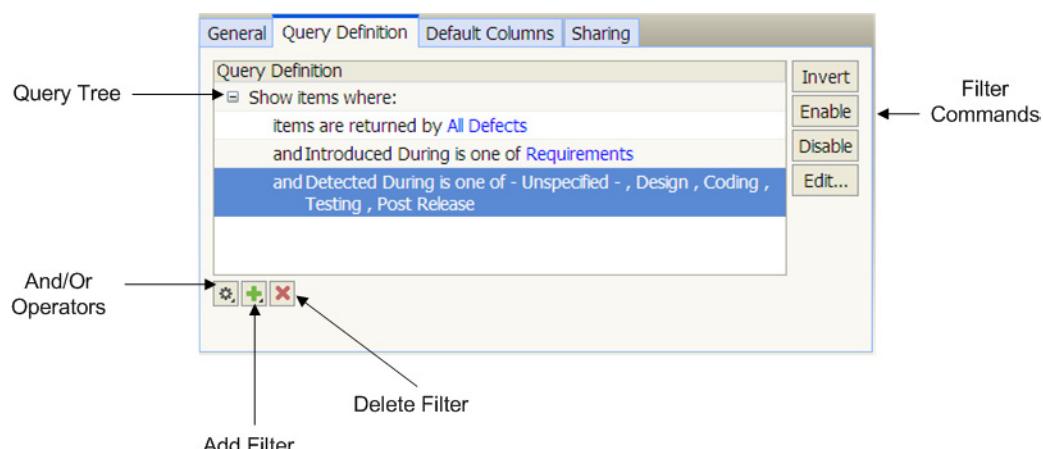
# Query Builder

The Query Builder allows you to create and edit queries as needed, and is available in the GUI and Web interface.

In the GUI, the Query Builder is available in the **New Query** and **Query Properties** dialog boxes, and can be displayed as a pane in the **Items** view. By default, the Query Builder pane displays with the Items view. If it is not visible, select **View > Settings**, and then select **Show query builder**. The following is an example of the **Query Builder** pane in the **Items** view:



In the Web interface, the Query Builder is available in the **Create Query** and **Query Properties** dialog boxes. The following is an example of a query in the **Create Query** dialog box:



To create a query, see “Creating a Query” on page 68.

# Query Filters

You determine what items are returned by a query by using *filters*. A filter limits the items returned by a query to those satisfying the conditions of the filter.

You can create a query using any combination of the following query filters:

**IMPORTANT** Your ability to use certain query filters is determined by the permissions granted to you to view specific item types and fields. For further assistance, contact your administrator.

Filter Type	Description
<b>Attachments</b>	<p>You can query for attachments that are linked to items with a specific field value or specific text.</p> <p>You can search the database for items that have an attachment or that have an attachment of a specified size, date or time added, name, or MIME (Multipurpose Internet Mail Extensions) type.</p> <p>Messages using the MIME standard can contain the following types of information: text messages in US-ASCII; character sets other than US-ASCII; multi-media, such as image, audio, and video messages; multiple objects in a single message; multi-font messages; messages of unlimited length; or binary files.</p>
<b>Branches</b>	<p>You can query on items that have a branch, or items that are branches of the original item.</p>
<b>Change Packages</b>	<p>If you use change packages to identify files that are affected by an item, you can track and monitor items with change packages using the change package filter.</p> <p>You can filter MKS Integrity items based on change package types and their attributes.</p> <p><b>Important:</b> Filtering on a change package attribute returns a list of all items that are associated with at least one change package that meets the specified criteria. There is currently no way to return only items where all of the associated change packages meet the specified criteria.</p> <p><b>Note:</b> If you use the same Implementer Change Package filter more than once in a query, the query assumes that they are grouped through a logical AND relationship. For example, if you used two <b>Member Name</b> filters in a query, one specifying <b>Order</b> and the other <b>Panel</b>, the query searches for items with change packages that have at least one member name containing <i>both</i> <b>Order</b> and <b>Panel</b>. It does not query items with change packages that have at least one member name containing <b>Order</b> and at least one member name containing <b>Panel</b>.</p>
<b>Items</b>	<p>You can create query filters to find content items used specifically in the document model. For example, you can create a query to search for segments, nodes, or for content defined as meaningful or non-meaningful content.</p> <p>You can also create query filters for test management items. Test management items are items that have a specific test management role, as defined by your administrator.</p>
<b>Fields</b>	<p>You can search the database for items whose fields contain specific values.</p>
<b>History</b>	<p>Query by history searches only the information in the change record. You may query by the following history filters:</p> <ul style="list-style-type: none"><li>■ specific value in an item field</li><li>■ date and time an item was modified</li><li>■ user who modified an item</li></ul> <p><b>Note:</b> If you use more than one history filter to search for specific values in multiple fields, the query searches for items with a history where all the fields were changed at the same time (that is, as part of the same edit).</p>
<b>Labels</b>	<p>You can query on labels by label name, or use a combination of label names to search all items. For example, you can query on <b>Defects</b> containing one of two different labels.</p>
<b>Relationships</b>	<p>You can query your database for items that are linked to other items. You can query for items with forward or backward relationships with other items linked through a specific relationship field. You can query for items that are linked to items with a specific field value or specific text.</p>

Filter Type	Description
<b>Sub-Query</b>	You can search the database for items that satisfy the conditions of a sub-query. <b>Note:</b> Using a sub-query in your new query does not alter the sub-query in any way.
<b>Test Results</b>	You can query your database for items that have a test result, that have a test result with an attachment or related item, or for a specific field value in the test result.
<b>Time Entries</b>	You can query your database for items that contain time entries. A time entry records the duration of time spent working on an item in hours for a specific day, and, optionally, any relevant notes.

## Grouping Query Filters

When adding a filter to a query, you group it with existing filters through a logical AND or OR relationship. A logical AND specifies that all of the conditions specified must be true for an item to be returned by the query. For example, an item must have both a `project = training` and `assigned group = docstaff` to be returned by the query. A logical OR specifies that one or more of the conditions must be true in order for an item to display in the query. For example, an item must either have a `state = submit` or a `priority not equal to high` to be returned by the query. Query filters are rendered in the query builder using simple text descriptions, for example:

```
Show items where:  
or where:  
    Project is one of /Financial Toolkit  
    and Assigned User is one of jriley  
or State is one of Submit  
or Priority is not one of High
```

# Creating a Query

---

**CLI EQUIVALENT** `im createquery`

---

You can create your own named queries to search for specific items.

## Example

To make sure that the developers can see all new and unassigned Change Request items, the project manager creates a shared query that returns all Change Requests in a state of `Submit`.

## Prerequisites

Before you create a query, ensure that you have the following:

- `CreateQuery` permission.
- Visibility permissions for the item types and fields you want to query on. Visibility rules restrict access to specific information based on project, and/or item type.

For further assistance, contact your administrator.

## To create a query in the GUI

- 1 Select **Query > Create**.
- 2 Provide information for the **Name** field, and the **Description**, **Image**, and **Sharing** tabs. For more information, see “Query Properties” on page 70.
- 3 Click the **Definition** tab, and then click **Add Filter** to add filters that determine what items display for the query.

By default, new filters are added to the root of the tree using an AND operator. If you want the new filter to have a logical OR relationship with an existing filter, select that filter and then select the `Or` option before clicking **Add Filter**.

To add a logical AND relationship between the new filter and an existing filter, select an existing filter, select the AND operator, and then click **Add Filter**.

To add a filter with a logical OR relationship between the new filter and an existing filter, select an existing filter, select the OR operator, and then click **Add Filter**.

---

**TIP** To clear a selected filter, hold the CTRL key while selecting that node.

---

For more information on grouping query filters, see “Grouping Query Filters” on page 67.

- 4 Use the drop-down lists and data filters to specify the types of filters and fields. For details on the types of filters, see “Query Filters” on page 66.

---

**TIP** For details on working with specific types of filters, see “Query Filters: Key Considerations” on page 70.

---

- 5 Where applicable, click **Next** to specify the options for the selected query filter. For details on query filter options, see “Query Filter Options” on page 73.
- 6 To add the filter to the query, click **Finish**.

- 
- 7 To customize the visible columns and sorting columns for the query, click the **Default Columns** tab. When you run the query, columns determine what item fields display in the Items view and how those fields are sorted. Customizing the default columns for a query is useful if you want to use different columns for different queries. For example, you could sort the list of items by State in ascending order.
  - 8 To save the query, click **OK**.

### To create a query in the Web interface

- 1 In the **Queries View**, click **Create Query**.
- 2 Provide information for the **General** and **Sharing** tabs. For more information, see “Query Properties” on page 70.
- 3 Click the **Query Definition** tab, and then click  to add the required filters to the query. From the list, choose one of the following:
  - To add the first filter or to add a new filter at the same level as an existing filter, select one of the following:
    - **Add a field value filter**
    - **Add a text search filter**
    - **Add a sub-query filter**
    - **Add a history filter**
    - **Other filters**
  - **Add a sub-expression** to add a sub filter to an existing filter. The line **and where** or **or where** is added under the selected filter. Select the sub-expression, and click  to add a filter. The **Filter Editor** dialog box displays.

The sub filter is added using the same operator as the main filter. To change the operator, select the sub filter, click , and select the new operator.

---

**NOTE** Any additional filters you add at the same level will be added with the same operator (**AND** or **OR**).

A sub-expression is a subset of criteria that applies to the parent filter. For example, if you had a filter that looked for two types of items, you could use a sub-expression for each to specify additional criteria. For example, if you were querying for Change Orders and Request For Change items, you could use a sub-expression on the Change Order filter to specify a Task change type and a sub-expression on the Request For Change filter to specify a Defect request type.

- 4 In the **Filter Editor** dialog box, define the filter criteria for the selected field or sub filter.

After you define a filter, it displays in the query definition.

If you do not define the filter, a generic filter line displays in the query definition with  to indicate that the definition is incomplete. Select the filter and click **Edit**. After you define the filter,  changes to  in the query definition.

- 5 To customize the visible columns and sorting columns for the query, click the **Default Columns** tab. When you run the query, columns determine what item fields display in the Items view and how those fields are sorted. Customizing the default columns is useful if you want to use different columns for different queries. For example, you could sort the list of items by State in ascending order.
- 6 To save the query, select **OK**.

## Query Properties

A named query contains the following properties:

Field/Tab	Description
<b>Name</b>	In the <b>Name</b> field, enter a name for the new query. Next to the <b>Name</b> field,  displays, indicating that the query is a <i>favorite</i> . Favorites are MKS Integrity objects (queries, charts, reports, and dashboards) that you created and use. To convert the query to an admin provided object, select <b>Is Admin Provided</b> . Admin provided objects are objects within the MKS Integrity object model that support solution definition and management, as well as workflow migration. For more information, see your administrator. <b>Note:</b> <ul style="list-style-type: none"><li>■ This option is available only if the <code>createSharedAdmin</code> or <code>Admin</code> permission is assigned to you.</li><li>■ Once a user object is converted to an admin provided object, you cannot revert it to a user object again.</li><li>■ Selecting the <b>Is Admin Provided</b> option automatically adds the currently logged in user to the <b>Sharing</b> tab with edit permissions.</li></ul>
<b>Description</b>	Description of the query, for example, Items assigned to the development group.
<b>Image (GUI only)</b>	Specifies an image for the query. Select <b>Use Custom Image</b> , <b>Default Image</b> , or <b>No Image</b> for your new query. If you are using a custom image, click <b>Select</b> , and browse to an image file. <b>Note:</b> Images must be GIF or JPEG format, and no larger than 16 by 24 pixels.
<b>Sharing</b>	Use the data filter to select the principals (users and groups) that you want to share your query with. <b>Note:</b> <ul style="list-style-type: none"><li>■ Only the groups you add to the <b>Shared With</b> list can see your query.</li><li>■ You cannot share a query to the <code>everyone</code> group unless you are an administrator or you have the <code>ShareToEveryone</code> permission. Contact your administrator for more information.</li></ul> To allow assigned principals to edit the query, select the principal's check box under the <b>Modify</b> column. To deny assigned principals the ability to edit the query, clear the principal's check box under the <b>Modify</b> column. By default, all assigned principals are denied the ability to edit a query.
<b>[Query] Definition</b>	Add the desired filters to create your query. For information on how to add a filter, see "Creating a Query" on page 68.
<b>Default Columns</b>	Customize the default columns for the query. For information on how to customize the default columns for a query, see "Creating a Query" on page 68.
<b>References (GUI only)</b>	Displays all admin provided and user objects that reference the existing query. If you plan on making changes to the query or deleting it, this information lets you know what objects are affected by the changes or deletion. For more information on admin provided objects, contact your administrator.
<b>History (GUI only)</b>	Displays all changes made to the existing query.

## Query Filters: Key Considerations

When using query filters in a query, consider the following:

Query Filter	Notes
General Query Information	<ul style="list-style-type: none"> <li>■ A named query can be edited by the user who created it. Principals (users and groups) that a named query is shared with can edit it if they have edit permissions assigned to them by the query creator. A named query can only be deleted by the user who created it or by the administrator.</li> <li>■ You must give the query a name, but you do not have to give it a description or share it with groups.</li> <li>■ To create or edit queries, you must have the <code>CreateQuery</code> permission assigned to you by your administrator.</li> <li>■ Queries with several filters may take longer to run than queries with only one or two filters.</li> <li>■ Your administrator defines states, ranges, and phases.</li> <li>■ Your administrator defines custom fields you can query on, for example, <code>Severity</code>, <code>Reviewed By</code>, <code>Root Cause</code>, <code>Lifecycle Phase</code>. When specifying the conditions for a custom field, if the field is not mandatory you can choose to query on an empty value.</li> <li>■ To include empty field values in your query, select the <b>or is empty</b> option. This option is only available for certain fields.</li> <li>■ Queries that include fields not visible on specific types do not return items where the fields are not visible.</li> </ul>
Attachment Filters	<p>When specifying conditions for attachment fields:</p> <ul style="list-style-type: none"> <li>■ Select the attachment field you want to query on from the data filter dialog box.</li> <li>■ Select the attachment attribute you want to query on.</li> <li>■ The list of fields you can select from is the list of fields in your database.</li> </ul>
Branch Filters	<ul style="list-style-type: none"> <li>■ If you do not have permission to view specific item types and/or their branches, branch filters do not display when modifying a query.</li> </ul>
Text Filters	<ul style="list-style-type: none"> <li>■ For text fields, type a word or phrase to search for <b>In the field name contains</b> field. Depending on the database, text field queries may be case sensitive (see your administrator for more information).</li> <li>■ To query on empty text fields, leave the <b>field name contains</b> field empty, and select the <b>or is empty</b> option.</li> <li>■ By default, MKS Integrity searches for items that contain all occurrences of words in a text string. For example, if you search for <code>print option</code>, MKS Integrity may only return items that contain <code>print</code> and <code>option</code> as separate words in a text field. To search for items that contain a complete text string, use double quotes, for example, “<code>print option</code>”.</li> <li>■ Because dynamically computed fields are not stored in the database, dynamically computed short text fields cannot be located with an all text field search in the MKS Integrity Client. To search for dynamically computed short text fields, create a query that includes a specific “<b>field contains</b>” comparison. For more information about computed fields, see your administrator.</li> <li>■ The <b>Any Text Field</b> filter is meant to return query results containing specific words in text fields, for example, “frequency”, “calculator”, and “hardware”. Common words, such as “the”, “this”, and “a”, are ignored; however, the filter in the Items view does accept common words, such as “the”.</li> <li>■ If the text specified for the <b>Any Text Field</b> filter is too generic, the query may stop because it consumes too many system resources.</li> </ul>

Query Filter	Notes
Date Filters	<ul style="list-style-type: none"> <li>■ For date fields, you can specify one of the following symbolic dates: Yesterday, Today, or Tomorrow. To set a specific date, select a date/time from the calendar when you click <b>Lookup Date</b>, or type a date/time using the MM/dd/yyyy HH:mm:ss format.</li> <li>■ When specifying a date range in a query, all dates are treated as timestamps, converted to the time zone on the MKS Integrity Server, and then truncated to a date-only format. The resulting date-only format is then converted to a Structured Query Language (SQL) statement format on the MKS Integrity Server, and the query is run based on the time zone of the server. If the user defining the date range is not in the same time zone as the MKS Integrity Server, a day can be lost or gained at either end of the defined range. The rollback of dates can cause the query results to vary.</li> </ul> <p><b>Important:</b> The date range conversion does not cause any problems when the user is in the same time zone as the MKS Integrity Server.</p> <p>For example, if the MKS Integrity Server is in the America/New_York time zone and the following query date range is defined by a user in Germany: Jan 1, 2009 to Jan 31, 2009.</p> <p>The final conversion of the date range is: Dec 31, 2008 06:00:00 AM GMT+01:00 to Jan 31, 2009 06:00:00 AM GMT+01:00.</p> <p>To avoid any date rollbacks when working with query date ranges in MKS Integrity, MKS recommends that users specify the time zone that is used by the MKS Integrity Server they are connecting to.</p> <ul style="list-style-type: none"> <li>■ If your administrator configured date fields to display the time, you can include the time in date filters by selecting the <b>Show Time</b> option and a corresponding time from the calendar. Time is specified from 00:00:00 to 23:59:59 inclusive in 24 hour format; however, MKS Integrity displays the time in 12 hour format. For example, specifying 13:56:45 displays the time as 1:56:45 PM.</li> <li>■ Displayed date fields do not change based on the time zone a user operates in; however, displayed date/time fields and time entries vary based on the time zone a user operates in.</li> <li>■ Symbolic dates are evaluated on the MKS Integrity Client's time zone.</li> </ul>
Label Filters	<ul style="list-style-type: none"> <li>■ If you do not have permission to view specific item types and those types have associated labels, label filters do not display when modifying a query.</li> <li>■ Searching on a blank label displays all items with labels you have permission to view.</li> </ul>
Change Package Filters	If you do not have permission to view any change package types, change package filters do not display when creating a query.
Item Filters	Item filters are used to find specific content and document item types that operate in a document model defined by your administrator.
Workflow and Document Project Filters	For workflow and document project fields, only projects you have permission to view are listed. Your administrator defines project permissions.
Configuration Management Project Filters	You cannot query on configuration management project fields.
Relationship Filters	<p>When specifying conditions for relationship fields:</p> <ul style="list-style-type: none"> <li>■ In the <b>on relationship</b> field, specify the relationship field that contains the related item.</li> <li>■ To restrict your query to either backward or forward relationships, the <b>backward</b> option is automatically selected if you select Backward Relationship in the <b>on relationship</b> field, and the <b>forward</b> option is automatically selected if you select Forward Relationships or a custom relationship field in the <b>on relationship</b> field. You cannot change these options.</li> <li>■ Relationship flags are set up by your administrator as part of the definition for a relationship field. The list of fields you can select from is the list of fields in your database.</li> </ul>
History (by value) Filters	You cannot query on attachment fields.

# Query Filter Options

The following is a detailed list of the available query filter options when defining a query:

## ***Query Filter Options: Attachments***

Option	Description
<b>Has Attachment</b>	Filters for items with at least one attachment.
<b>File Name</b>	Filters for items with attachment of specific name (when specifying conditions for this field, enter attachment's file name with or without extension).
<b>File Size</b>	Filters for items with attachment file of specific size.
<b>MIME Type</b>	Filters for items with attachment of specific MIME type.
<b>Summary</b>	Filters for items with a summary if one was added.
<b>Date Added</b>	Filters for items with attachments created on a specific date.
<b>Created By</b>	Filters for items with attachments created by specific users.

## ***Query Filter Options: Items***

Option	Description
<b>Is a document node</b>	Filters for nodes in a document.
<b>Is a document</b>	Filters for segments.
<b>Is a Shared Item</b>	Filters for shared items that are referenced by nodes.
<b>Is a content node</b>	Filters for nodes representing content.
<b>Is a sub-segment node</b>	Filters for nodes representing subsegments.
<b>Content is meaningful</b>	Filters for categories previously defined as meaningful content. For example, a managed artifact is meaningful if it is not a Comment or a Heading.
<b>Content is non-meaningful</b>	Filters for categories previously defined as non-meaningful content. For example, a Comment or a Heading.
<b>Is a group document</b>	Filters for documents that are used to group test cases for text execution.
<b>Is a test step</b>	Filters for items that have a test management role of test step.
<b>Is a test case</b>	Filters for items that have a test management role of test case.
<b>Is a test session</b>	Filters for items that have a test management role of test session.
<b>Is a test suite</b>	Filters for items that have a test management role of test suite.

## ***Query Filter Options: Branches***

Option	Description
<b>Has a branch</b>	Filters for originating items that have been branched.
<b>Is a branch</b>	Filters for items that are a branch of an originating item.

## ***Query Filter Options: History***

Option	Description
<b>Assigned Group</b>	Filters for items with changes to assigned group in item history.
<b>Assigned User</b>	Filters for items with changes to assigned user in item history.
<b>Project</b>	Filters for items with changes to item's project in item history.

Option	Description
<b>Signature Comment</b>	Filters for items with changes to signature comment in item history.
<b>Signed By</b>	Filters for items with changes to user who signed item change in item history.
<b>State</b>	Filters for items with changes to item state in item history.
<b>Summary</b>	Filters for items with changes to item summary in item history.

#### **Query Filter Options: Labels**

Option	Description
<b>Labels</b>	Filters for items with the specified label name.

#### **Query Filter Options: Configuration Management Change Packages**

Option	Description
<b>Has Change Package</b>	Filters for items with change packages included in at least one generic change package, such as Implementer.
<b>Has Resolution Change Package</b>	Filters for items with change packages included in at least one MKS Integrity change package.
<b>Closed Date</b>	Filters for items with change packages closed on specific date.
<b>Created By</b>	Filters for items with change packages created by specific user.
<b>Created Date</b>	Filters for items with change packages created on specific date.
<b>Deploy Request ID</b>	Filters for items with change packages that contain specific deploy request ID. (To use this filter, you must be licensed for Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .)
<b>Deploy Request State</b>	Filters for items with change packages associated with deploy requests in specific state. (To use this filter, you must be licensed for Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .)
<b>Deploy Target</b>	Filters for items with change packages that contain specific deploy target name. (To use this filter, you must be licensed for Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .)
<b>Description</b>	Filters for items with change packages that contain specific description.
<b>Item ID</b>	Filters for items by item ID.
<b>Subtype</b>	Filters for items with change packages of specific type. (To use this filter, you must be licensed for Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .)
<b>Server Hostname</b>	Filters for items with change packages on specific server name (use this field with <b>Server Port</b> field).
<b>Server Port</b>	Filters for items with change packages on specific server port (use this field with <b>Server Hostname</b> field).
<b>Stage</b>	Filters for items with change packages associated with specific stage. (To use this filter, you must be licensed for Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .)
<b>Staging System</b>	Filters for items with change packages associated with specific staging system name. (To use this filter, you must be licensed for Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .)
<b>State</b>	Filters for items with change packages in specific state.
<b>Summary</b>	Filters for items with change packages that contain specific summary.

## Query Filter Options: Configuration Management Change Packages Entry Attributes

Option	Description
<b>Archive</b>	Filters for items with change package entries with specific archive.
<b>Bytes Added</b>	Filters for items with change package entries that contain specific number of bytes added by member operation (use for binary archives).
<b>Bytes Deleted</b>	Filters for items with change package entries that contain specific number of bytes deleted by member operation (use for binary archives).
<b>Date Changed</b>	Filters for items with change package entries modified on specific date.
<b>Hostname</b>	Filters for items with change package entries on specific server name (use this field with Port field).
<b>Lines Added</b>	Filters for items with change package entries that contain specific number of lines added by member operation (use for text archives).
<b>Lines Deleted</b>	Filters for items with change package entries that contain specific number of lines deleted by member operation (use for text archives).
<b>Location</b>	Filters for items with change package entries containing specific backing archive for member, or a backing project for a subproject.
<b>Member</b>	Filters for items with change package entries that contain specific member name.
<b>Member Type</b>	Filters for items with change package entries containing specific project element affected by operation (available options are Subproject, Member, or Unknown).
<b>Modifier</b>	Filters for items with change package entries with specified modifier of either Committed or Pending.
<b>Port</b>	Filters for items with change package entries on specific server port (use this field with Hostname field).
<b>Project</b>	Filters for items with change package entries that belong to specific MKS Integrity project (when specifying conditions for this field, type project name in text field, for example, c:/masterdemo/project.pj; typing path to your project is optional).
<b>Revision</b>	Filters for items with change package entries that contain members with specific revision number.
<b>Text</b>	Filters for items with change package entries that contain text or binary archives (available options are true (text) or false (binary)).
<b>Type</b>	Filters for items with change package entries that added members or subprojects by specific operation.
<b>Variant</b>	Filters for items with change package entries on specific MKS Integrity development path.

## Query Filter Options: Relationships

Option	Description
<b>Has Relationship</b>	<p>Filters for items with at least one related item.</p> <p>When specifying conditions for relationship fields:</p> <ul style="list-style-type: none"> <li>■ In <b>on relationship</b> field, specify relationship field that contains related item.</li> <li>■ To restrict your query to either backward or forward relationships, <b>backward</b> option is automatically selected if you select Backward Relationship in <b>on relationship</b> field, and <b>forward</b> option is automatically selected if you select Forward Relationships or custom relationship fields in <b>on relationship</b> field. You cannot change these options.</li> </ul> <p>Your administrator sets up Relationship flags as part of definition for relationship field. List of fields you can select from is list of fields in your database.</p>

Option	Description
<b>Test any flag <i>flagname</i> on field <i>fieldname</i> is set</b>	Filters for items with specific relationship flag set on specific relationship field. <b>Note:</b> If the field you select does not have a relationship flag defined you are not able to complete the filter definition. Use the Back button to make another selection.
<b>Fields</b>	Filters for items with related item with a specific field value or specific text.

### **Query Filter Options: Time Entries**

Option	Description
<b>Has Time Entry</b>	Filters for items with at least one time entry.
<b>Created Date</b>	Filters for items with time entries created on specific date.
<b>Created By</b>	Filters for items with time entries created by specific user.
<b>Modified Date</b>	Filters for items with time entries modified on specific date.
<b>Modified By</b>	Filters for items with time entries modified by specific user.
<b>Item ID</b>	Filters for items with time entries that contain specific item ID.
<b>User</b>	Filters for items with time entries that contain specific user.
<b>Entry Date</b>	Filters for items with time entries that contain specific entry date.
<b>Duration</b>	Filters for items with time entries that contain specific duration period in hours.
<b>Notes</b>	Filters for items with time entries that contain specific text in <b>Notes</b> field.
<b>Source</b>	Filters for items with time entries created by a specific source (available options is mks_integrity).

### **Query Filter Options: Test Result**

Option	Description
<b>Is Related to By a Test Result</b>	Filters for items that are related to at least one test result.
<b>Has Test Result</b>	Filters for items with at least one test result.
<b>Has Test Result With Related Item</b>	Filters for items with at least one test result that has a related item.
<b>Has Test Result With Attachment</b>	Filters for items with at least one test result that has an attachment.
<b>Test Result &lt;field&gt;</b>	Filters for items with at least one test result that matches the specified field value. The field can be one of the following: Modified Date, Session ID, Modified By, Verdict, or Verdict Type.

# Editing a Query

---

**CLI EQUIVALENT** im editquery

---

You can edit the Quick Query or any named queries you create. Principals (users and groups) a named query is shared with can edit it if they have edit permissions assigned to them by the query creator.

## **Prerequisites**

Before you edit a query, ensure that you have the following:

- CreateQuery permission.
- Visibility permissions for the item types and fields you want to query on. Visibility rules restrict access to specific information based on project, and/or item type.

For further assistance, contact your administrator.

## **Key Considerations**

- You cannot hide, delete, share with other users and groups, rename, or add a description/image to the Quick Query.
- Queries with several filters may take longer to run than queries with only one or two filters.
- You can edit a query's default columns if they are currently being used in an Items view; you are prompted to run the query to update the Items view after you make the changes.
- You cannot save changes to a query that is an admin provided object. You must save it as a copy first.

## **To edit a query**

Interface	Procedure
GUI	Select a query from the <b>Manage Queries</b> view or run a query, and select <b>Query &gt; Edit</b> .
Web	From the <b>Queries</b> view, select a query and select <b>Query Properties</b> .

# Running a Query

---

## CLI EQUIVALENT im issues

---

The first time you log in to MKS Integrity, the Quick Query runs by default in the Items view. To view a list of items generated by another query, you must run the desired query.

### Prerequisites

Before you run a query, ensure that you have the following:

- CreateQuery permission.
- Visibility permissions for the item types and fields you want to query on. Visibility rules restrict access to specific information based on project and/or item type.

For further assistance, contact your administrator.

### Key Considerations

- It is possible to create complex queries that demand significant system resources when run against a large database of items. Queries that return a large result count can also demand significant memory resources from both the MKS Integrity Server and the MKS Integrity Client. To address this item, your MKS Integrity administrator may decide to limit the returned item count or set query timeouts for certain groups of users.

If such query limits are specified and you run a query that exceeds those limits, MKS Integrity displays an error message advising that the set limit has been exceeded. If this is a consistent problem with your queries, contact your administrator.

- When you open and close the GUI or create a new instance of the Items view, the last run query displays.
- In the GUI, you can display recently run queries by adding the **used recently** pre-defined filter in the **select query to run** list.

### To run a query

Interface	Procedure
GUI	In the <b>Items</b> view, select a query from the <b>Query</b> list. or In the <b>Manage Queries</b> view, select a query and select <b>Query &gt; Run</b> .
Web	In the <b>Items View</b> , select query from the <b>Query</b> list. or In the <b>Queries View</b> , select a query and click <b>Run Query</b> .

After you run a query, the query results display in the Items view. Any filters currently associated with the Items view are applied in addition to the query filters. If no filters are currently associated with the Items view, you can apply them to refine the query results. For more information on filtering, see “Filtering Items (Web only)” on page 18.

# Copying a Query

---

**CLI EQUIVALENT** `im copyquery`

---

You can copy a query and rename the copy to create a new query. This is useful when you want to create a query that is similar to an existing complex query. Copying the Quick Query creates a named query that can be shared with others.

## Prerequisites

Before you copy a query, ensure that you have the following:

- `CreateQuery` permission
- Visibility permissions for the item types and fields you want to query on. Visibility rules restrict access to specific information based on project and/or item type.

For further assistance, contact your administrator.

## Key Considerations

- By default, the original query definition is referenced in the new query as a subquery so that any changes to the original query definition are reflected in the new query. If you want to create a static copy of the query, do the following on the **Definition** tab:
  - In the GUI, select the **Copy Query** option.
  - In the Web interface, select **Set definition to copy the original query**.
- A copied favorite (★) or non-favorite (☆) query becomes a favorite when you save the new query.
- Sharing information is not copied. You can copy sharing information by clicking **Copy Principals** on the **Sharing** tab.
- You cannot share a query to the `everyone` group unless you are an administrator or you have the `ShareToEveryone` permission. For more information, contact your administrator.

## To copy a query

Interface	Procedure
GUI	From the <b>Manage Queries</b> view, select a query, and select <b>Query &gt; Copy</b> .
Web	From the <b>Queries View</b> , select a query and click <b>Copy Query</b> .

---

# Deleting a Query

---

**CLI EQUIVALENT** `im deletequery`

---

Eventually, you may want to delete named queries you no longer use, or if you have too many to manage.

You cannot delete the Quick Query; however, you can clear it, resetting the default fields to empty values.

## **Prerequisites**

Before you delete a query, ensure that you have the `CreateQuery` permission and are the query owner. Only the query owner or an administrator can delete named queries.

For further assistance, contact your administrator.

## **To delete a query**

Interface	Procedure
GUI	From the <b>Manage Queries</b> view, select a query, and select <b>Query &gt; Delete</b> .
Web	From the <b>Queries View</b> , select a query and click <b>Delete Query</b> .

---

# Exporting Query Results to Microsoft Excel

---

## CLI EQUIVALENT im exportissues

---

You can export the visible item fields and columns returned by a query to Microsoft Excel from the Items and Relationships (GUI and Web), and Document views (GUI only) for data analysis. From a historical view, you can also export items as of a historical date or label.

---

**IMPORTANT** Exporting large query result sets may affect MKS Integrity Server performance.

---

When you export items, MKS Integrity saves the output as a temporary Excel file (.xls) that you can rename and save.

- In the Windows GUI, the temporary Excel file automatically opens in Microsoft Excel.
- In the UNIX GUI, a dialog box indicates the path and name of the temporary Excel file.
- In the Web, the temporary Excel file is downloaded by your browser.

After the temporary file is open in Excel, you can then manipulate the data to create charts and reports for further analysis.

---

**IMPORTANT** After data is exported to Excel, you cannot import the data back into MKS Integrity. Exporting query results to Excel is intended for data analysis only. To export and import data between MKS Integrity and Excel, use the Microsoft Excel integration with MKS Integrity.

---

## Key Considerations

- MKS Integrity supports exporting items to Microsoft Excel 2003, 2007, and 2010.
- Excel has the most current data as retrieved from the server but the data in the MKS Integrity Client may be out of synch with the data retrieved from the server. Refresh the client to ensure you are viewing the most current data.
- Sort order and filters applied to the view are maintained in the exported data.
- If parameter substitutions are enabled in MKS Integrity, they will be visible in the Excel output.
- If you exit the MKS Integrity Client on Windows while the temp file is still open in Excel, you are still able to save the file.
- Before you export query results, note that Excel contains the following data limitations:
  - maximum 32,767 characters per cell
  - maximum 65,536 rows per sheet
  - maximum 256 columns per sheet

When the row limit is exceeded, data is split across sheets.

Long text fields that exceed the character limitation per cell are truncated, with "[Truncated]" appended to the end of the cell.

For a complete list of Excel specifications and limitations, browse to:

<http://office.microsoft.com/en-us/excel-help/excel-specifications-and-limits-HP005199291.aspx>

- 
- MKS Integrity's date/time fields format (based on your current locale) is not supported in Excel. Instead, Excel uses MM/dd/yyyy for date fields and MM/dd/yyyy HH:mm:ss for date/time fields. For example, the date/time field Jul 22, 2009 5:15:14 AM in MKS Integrity displays as 07/22/2009 05:15:14 in Excel.
  - The following MKS Integrity display patterns are supported in Excel:
    - "#,###"
    - "#,###.##"
    - *currency symbol* + "#,###"
    - *currency symbol* + "#,###.00"
    - "#,###;(#,###)"
    - "#,###.##;(#,###.##)"
    - "0.###E0" (displays as "0.###E+0" in Excel)
    - "#%"

MKS Integrity display patterns that use exclusive text are not supported in Excel. Any unsupported display pattern is disregarded; however, a standard display pattern of "0" for integer and "0.00" for floating point fields is applied.

- In the GUI and the Web, column widths from the view specify column widths in Excel. In the CLI, default column widths specify column widths in Excel.
- If wrapping is enabled for table content in MKS Integrity, cell content is wrapped in Excel.
- If no items are exported, Excel does not start, for example, if you attempt to export an item that you do not have permission to view.
- In the Relationships view, the **Relationship Flags** and **Order** fields are not exported. In the Document view, the **Section** field is not exported.
- The following describes the format of each MKS Integrity field after exporting to Excel:
  - ID fields display as a number (0 decimal places, no other formatting).
  - Floating-point and integer fields display as numbers with display patterns (see above for examples).
  - State, type, project, phase, range, SI Project, short text, long text, user, logical, and FVA fields display as text.

---

**NOTE** Rich content formatting does not display on long text fields.

---

- Attachment fields display as a CSV list of file names.
- Relationship fields display as a CSV list of ID/relationship flags.
- Query Backed Relationship (QBR) fields display as a CSV list of IDs.
- IBPL fields display as a CSV list of the field(s) visible in a text format.
- Group and pick fields display as a CSV list of text.
- Date fields display as a date.

## To export query results to Microsoft Excel from the GUI or the Web

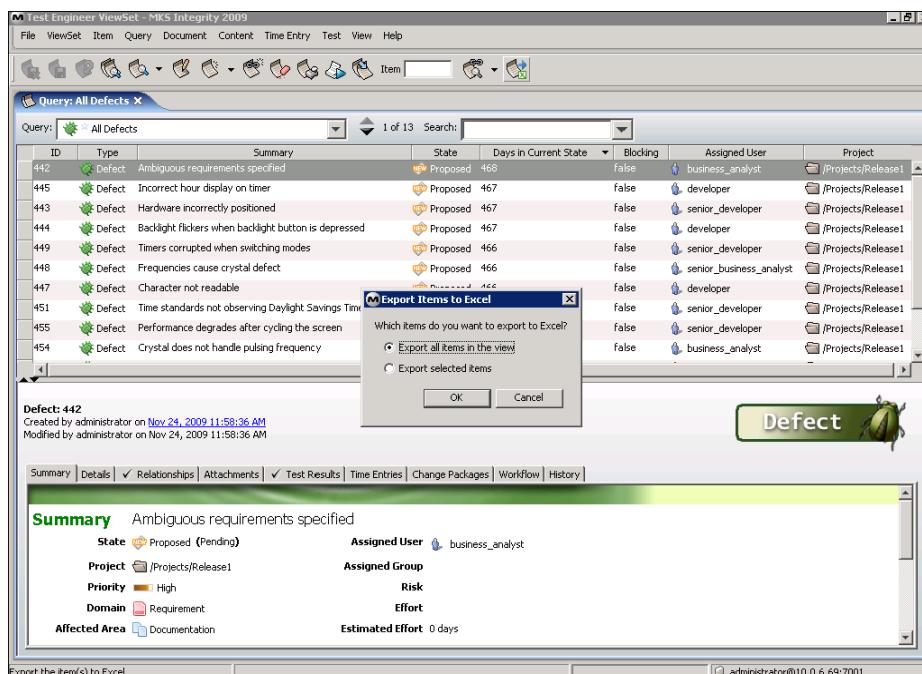
1 From the Items, Relationships, or Document view (GUI only), do one of the following:

- To export specific items to Excel, select the items and select **Export Items to Excel**.
- To export all items to Excel, select **Export Items to Excel**.

A dialog box prompts you to choose to either export the selected items, or all items to Excel. By default, **Export all items** in the view is selected.

2 Click **OK**.

GUI:



Web:

ID	Type	Summary	State	Assigned User	Project
108	Work Item	Timezone +13	Completed	dtashnev	/Projects/Release1
109	Work Item	Timezone +12 (UTC coordinated)	Completed	dtashnev	/Projects/Release1
110	Work Item	Timezone +10 (Vladivostok)	Completed	dtashnev	/Projects/Release1
112	Work Item	Timezone -10	Completed	dtashnev	/Projects/Release1
113	Work Item	System and interface testing	In Testing	dtashnev	/Projects/Release1
114	Work Item	Feature testing	In Testing	dtashnev	/Projects/Release1
115	Work Item	Daylight Savings Time scenario tests	In Testing	dtashnev	/Projects/Release1
116	Work Item	Accuracy of Chronometer	In Testing	dtashnev	/Projects/Release1
401	Work Item	Alarm Clock notification and reliability	In Testing	dtashnev	/Projects/Release1
402	Work Item	Visual implementation testing	In Testing	dtashnev	/Projects/Release1
445	Work Item	Interaction between navigation modes	In Testing	dtashnev	/Projects/Release1
615	Test Objective	Weekly regression testing results	In Testing	dtashnev	/Projects/Release1
616	Test Objective	System and interface testing	In Testing	dtashnev	/Projects/Release1
617	Test Objective	Feature testing	In Testing	dtashnev	/Projects/Release1
618	Test Objective	Daylight Savings Time scenario tests	In Testing	dtashnev	/Projects/Release1
619	Test Objective	Accuracy of Chronometer	In Testing	dtashnev	/Projects/Release1
620	Test Objective	Alarm Clock notification and reliability	In Testing	dtashnev	/Projects/Release1
621	Test Objective	Visual implementation testing	In Testing	dtashnev	/Projects/Release1
623	Test Objective	Interaction between navigation modes	In Testing	dtashnev	/Projects/Release1
816	Test Objective	Weekly regression testing results	In Testing	dtashnev	/Projects/Release1
817	Test Objective	System and interface testing	In Testing	dtashnev	/Projects/Release2
818	Test Objective	Feature testing	In Testing	dtashnev	/Projects/Release2
819	Test Objective	Regression testing	In Testing	dtashnev	/Projects/Release2

In Excel, the column names display in the first row, and data displays in the following rows:

A	B	C	D	E	F	G	H
ID	Type	Summary	State	Days in Current State	Blocking	Assigned User	Project
1	442	Defect Ambiguous requirements specified	Proposed	468	false	business_analyst	/Projects/Release1
2	445	Defect Incorrect hour display on timer	Proposed	467	false	developer	/Projects/Release1
3	443	Defect Hardware incorrectly positioned	Proposed	467	false	senior_developer	/Projects/Release1
5	444	Defect Backlight flickers when backlight button is depressed	Proposed	467	false	developer	/Projects/Release1
6	449	Defect Timers corrupted when switching modes	Proposed	466	false	senior_developer	/Projects/Release1
7	448	Defect Frequencies cause crystal defect	Proposed	466	false	senior_business_analyst	/Projects/Release1
8	447	Defect Character not readable	Proposed	466	false	developer	/Projects/Release1
9	451	Defect Time standards not observing Daylight Savings Time	Proposed	465	false	senior_developer	/Projects/Release1
10	455	Defect Performance degrades after cycling the screen	Proposed	459	false	senior_developer	/Projects/Release1
11	454	Defect Crystal does not handle pulsing frequency	Proposed	459	false	business_analyst	/Projects/Release1
12	459	Defect Timezone screens do not match	Proposed	457	false	developer	/Projects/Release1
13	458	Defect Improperly formatted chronometer layout	Proposed	457	false	developer	/Projects/Release1
14	461	Defect Improper display on the chronometer screen	Proposed	456	false	developer	/Projects/Release1
15							

## PART 3

# Displaying Item Data Graphically

---

# What Is a Chart?

A *chart* is a summary of item data presented in a graphical format. When you create a chart, you can restrict what data is included based on the values in standard or custom item fields. For example, you may want to see only the data for items with a **Type** field value of **Defect**.

You can create charts using one of four chart types: trend chart, distribution chart, item fields chart, or item fields trend chart.

A *trend chart* reports item data over a specified period of time summarized at specific intervals. For example, you may want to see the number of defects created over the last month, summarized on a weekly basis. You also have the option of restricting the data to items belonging to a specific project or projects. For example, you may want to see only the number of defects created over the last month for the **Cosmos** project.

A *distribution chart* reports item data for items based on an existing query. For example, you may want to base your chart on a query that shows open items for a specific project in order to see the current number of defects for that project. You can then design the chart to break down the current number of defects based on a specified field value. For example, you may want to see the current number of defects broken down by state.

In a distribution chart, you can also perform aggregate operations against numeric fields in items returned by an existing query. *Aggregate operations* allow you to add numeric field values, calculate the average of multiple numeric field values, retrieve the smallest or largest numeric field value in a group of items, or count the number of items returned by an existing query. For example, in addition to creating a distribution chart that groups several Project items by state, you can also add the **Budget** field in each state grouping to display a total budget by state. For more information on computed expressions, see “What are Computed Expressions?” on page 159.

An *item fields chart* reports numeric field data from items based on an existing query and a computed expression. The computed expression determines chart values by performing arithmetic calculations between numeric fields in each item returned by the query. For example, you may want to base your chart on a query that shows active Project items and create a computed expression that calculates the discrepancy between the **Estimated Cost** and **Actual Cost** fields in each Project item returned by the query.

An *item fields trend chart* reports numeric field values over a specified period of time summarized at specific intervals. For example, an item fields trend chart could report on the changing values of the **Planned Effort** field for several Project items over a period of several months.

Charts can be displayed in a number of different graph styles. You can choose from line, bar, pie, XY (scatter), or bubble formats. You can customize the colors, legend, and orientation of the graph, and you can make bar and pie graphs three-dimensional. You can also display chart data in table format.

If you are reporting on a large number of field values, you can use aliases to group the data under a broader category. For example, if you are reporting on all possible states for an item, you could assign an alias of **In Work** to both the **In Progress** and **In Development** field values. You can also use aliases to give a single field value a more meaningful name.

---

# Charts View: Managing Charts

---

**CLI EQUIVALENT** `im charts`

---

You can view and manage all charts created by you or shared to you from the **Manage Charts** view (GUI) or **Charts View** (Web interface).

To display the **Manage Charts** view in the GUI, select **Charts > Manage Charts**.

To display the **Charts View** in the Web interface, under **Quick Views**, select **Charts**.

The data filter in the **Charts** view displays your favorite charts (charts that you created and use) by default. You can search for a specific chart by typing in the text filter and/or specifying a filter.

In this view, you can:

- create a chart (GUI only) (see “Creating a Chart” on page 89)
- edit a chart (GUI only) (see “Editing a Chart” on page 109)
- run a chart (see “Running a Chart” on page 110)
- delete a chart (see “Deleting a Chart” on page 114)
- copy a chart (GUI only) (see “Copying a Chart” on page 112)
- save and print a chart (see “Saving and Printing a Chart” on page 113)

---

# Chart Definition View

---

**CLI COMMAND:** `im viewchart`

---

The **Chart Definition** view displays the details of a chart definition.

---

**NOTE** You cannot edit the chart definition while viewing it, and you cannot display a chart definition in the Web interface.

---

To display this view, select a chart in the **Manage Charts** view, and select **Chart > View Definition**.

To view specific information about the chart, click a tab.

---

# Creating a Chart

---

**CLI EQUIVALENT** `im createchart`

---

In the GUI, you can create four types of charts:

- distribution chart (see “To create a distribution chart” on page 90)
- item fields chart (see “To create an item fields chart” on page 91)
- trend chart (see “To create a trend chart” on page 92)
- item fields trend chart (see “To create an item fields trend chart” on page 94)

## ***Editing Restrictions***

- The user who created the chart can edit it. Principals (users and groups) that a chart is shared with can edit it if they have edit permissions assigned to them by the chart creator. Only the user who created the chart or an administrator can delete the chart.
- While creating a chart, you cannot edit the query that the chart is based on.
- All charts are subject to visibility rules set by your administrator. Visibility rules restrict access to specific information based on project and/or item type. For more information, contact your administrator.

## ***Minimum Information Required***

- To create a distribution chart, you require a chart name, a field, and a query.
- To create a trend chart, you require a chart name, step type, start and end dates, and a field.
- To create an item fields chart, you require a chart name, query, and aggregate expression.
- To create an item fields trend chart, you require a chart name, query, step type, start and end dates, and numeric field.
- All other modifications and additional information are optional.

## ***Calculations in Charts***

Charts can do more than just display field information in a graphical format. You can also perform arithmetic calculations between numeric fields and display the values in the chart. For example, you can calculate the average for a group of field values or count the number of items in a specific state. To perform these calculations, you create a *computed expression*. For more information, see “What are Computed Expressions?” on page 159.

## ***How Time Zones Affect Charts***

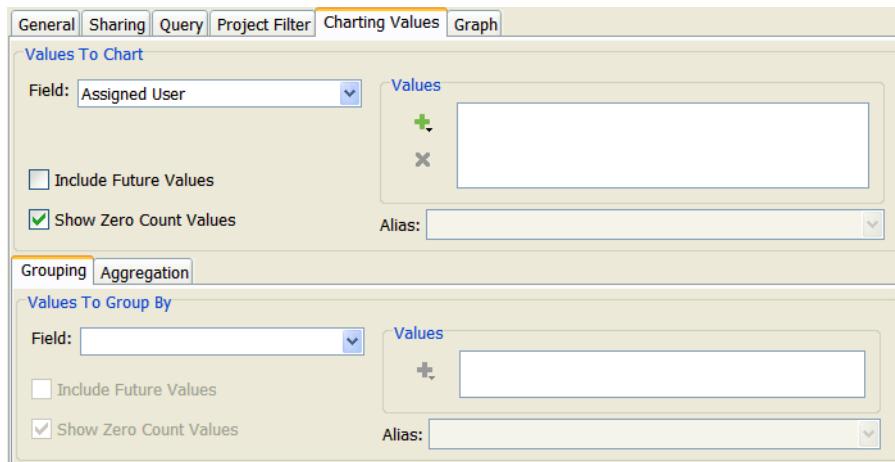
- Symbolic dates in rules and queries are evaluated in the MKS Integrity Client’s time zone.
- Relevance and editability rules are evaluated in the MKS Integrity Client’s time zone.
- Computed expressions return dates/times in the MKS Integrity Client’s time zone and perform calculations in the MKS Integrity Server’s time zone where appropriate.

## Examples

For details on how to define the key elements of sample charts of each type, see “Detailed Chart Examples” on page 99. For an example of how to create a more complex chart using computed fields, see “Creating a Computed Expression in a Chart” on page 179.

### To create a distribution chart

- 1 Select **Chart > Create**. The **Create Chart** dialog box displays the **General** panel.
- 2 Provide information for the **General**, **Sharing**, **Query**, and **Project Filter** tabs. For detailed information, see “Chart Dialog Box: General, Sharing, Query, and Project Filter Tabs” on page 104.
- 3 Click the **Charting Values** tab. The **Charting Values** panel displays.



- 4 Under **Values to Chart**, click the **Field** list and select the field you want to report on. For example, if you select **Type**, chart data is based on one or more values in the **Type** field.
- 5 Under **Values**, select the field value(s) to chart.
- 6 If you want the field value to display in the chart with a different name, select a field value in the **Values** list, then type or select the new name in the **Alias** field. If you want multiple field values to display on the chart combined under one name, select the field values in the **Values** list, then select or type a name in the **Alias** field. For example, if you are reporting on **Docs**, **Feature Request**, and **Defect** item types, and you assign an alias of **Development** to both **Feature Request** and **Defect** field values, the data for these item types is combined in the chart under **Development**.
- 7 To include in the chart automatically any new values that are defined for the field in the future, select **Include Future Values**.
- 8 If you do not want to include field values in the chart if there are no items with that value, deselect **Show Zero Count Values**.

#### NOTE

- To hide zero count values in a distribution pie chart, all values for the field being charted must equal zero. For example, if you are charting the **Type** and **State** fields and the **Defect** type has zero items in the **Submit** state and the **Feature** type has 10 items in the **Submit** state, the **Defect** type displays a value of 0 in the chart.
- If more than one chart is produced by a single chart definition (through the use of grouping values), to hide zero count values, the total values for the field being charted must equal zero across all charts.

- 9 In the **Grouping** tab under the **Values to Group By**, click the **Field** list and select a field to break down the data in the chart. For example, if you select **State** as your grouping field, a bar graph is

---

broken down into separate bars based on the value in the **State** field; however, a stacked bar graph is broken down into groups of bars by state. The list of field values changes to display the available values for the field you select.

---

**NOTE** Once you specify grouping values, a check mark displays on the **Grouping** tab. This allows you to see that the chart contains grouping values without having to click the tab.

- 10 Under **Values**, select the field value(s) to group. For example, if you are charting the **Type** field and you select Feature Request and Defect, the chart only shows data for items that have either of these values in the **Type** field.
- 11 Specify an alias for a field value in the **Values** list.
- 12 To break down the chart data automatically for any new values that are defined for the grouping field in the future, select **Include Future Values**.
- 13 If you do not want to include a breakdown in the chart for grouping field values, if there are no items with that value, deselect **Show Zero Count Values**.
- 14 To perform an aggregate computation against a numeric field included in the chart and display the result in the chart, click the **Aggregation** tab. *Aggregate computations* allow you to add numeric field values, calculate the average of multiple numeric field values, retrieve the smallest or largest numeric field value in a group of items, or count the number of items returned by an existing query. The **Aggregation** panel displays the aggregate expression `count()` by default. This expression returns the number of items in the specified query. Once you create another expression, you can delete the `count()` expression; however, a distribution chart must have at least one aggregate computation.
- 15 To add an aggregate expression, click **Add**. The **Add Expression** dialog box displays. For more information, see “Distribution Chart: Add/Edit Expression Dialog Box” on page 106.  
To edit an existing aggregate expression, select it, and click **Edit**. The **Edit Expression** dialog box displays. For more information, see “Distribution Chart: Add/Edit Expression Dialog Box” on page 106.  
If there are multiple aggregate expressions in the **Aggregation** panel, you can change the order of an aggregate expression by selecting it and clicking **Move Up** or **Move Down**.  
To remove an aggregate expression from the chart, select it, and click **Remove**.

---

**NOTE** If there is only one aggregate expression, **Remove** is unavailable.

- 16 To specify a graph style in the **Graph** panel, see “To specify a graph for a chart” on page 96.
- 17 To finish and save your chart, click **OK**.

### To create an item fields chart

- 1 Select **Chart > Create**. The **Create Chart** dialog box.
- 2 From the **Chart Type** list, select **Item Fields**.
- 3 Provide information for the **General**, **Sharing**, **Query**, and **Project Filter** tabs. For detailed information, see “Chart Dialog Box: General, Sharing, Query, and Project Filter Tabs” on page 104.
- 4 Click the **Charting Values** tab. The **Charting Values** panel displays.
- 5 To perform an arithmetic calculation between numeric fields included in the chart and display the result in the chart, click **Add**. The **Add Expression** dialog box displays. For more information, see “Item Fields Chart: Add/Edit Expression Dialog Box” on page 107.

**TIP** Double-clicking the empty expression table also displays the **Add Expression** dialog box.

To edit an existing computed expression, select it and double-click, or click **Edit**. The **Edit Expression** dialog box displays. For more information, see “Item Fields Chart: Add/Edit Expression Dialog Box” on page 107.

If there are multiple computed expressions in the **Charting Values** panel, you can change the order of a computed expression by selecting it and clicking **Move Up** or **Move Down**.

To remove a computed expression from the chart, select it, and click **Remove**.

- 6 In the **Item Identifier** field, specify the field you want to identify items by in the chart. For example, if you type {ID} each item in the chart is identified by the value of the **ID** field. An item identifier displays as an axis label in the chart. For better visibility, you should keep item identifiers short. You can also manually insert fields into the **Item Identifier** field by selecting a field from the field list and clicking **Insert Field**.

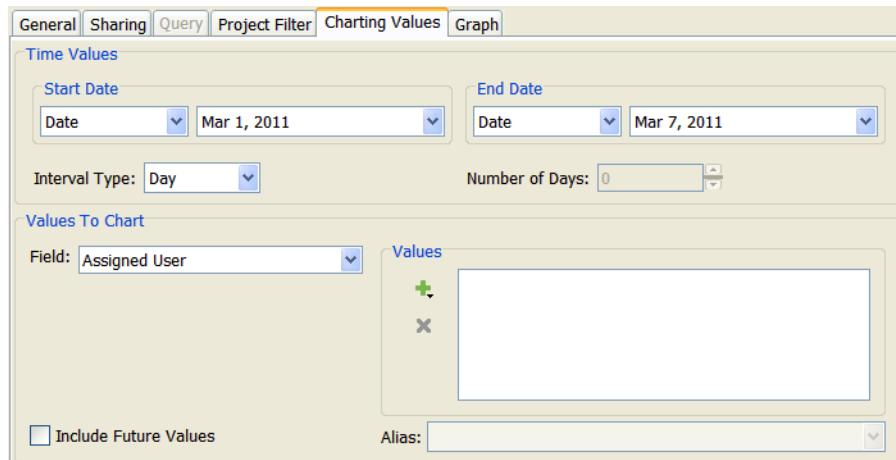
If you want to add text that precedes the field, type it before the field, for example, `Project: {ID}`. The chart then identifies each item by displaying `Project: ID field value`. You can specify multiple fields, for example, `Project: {ID} {State} {Priority}`.

**NOTE** Because item identifiers display as axis labels in the chart, you should keep identifiers short.

- 7 To specify a graph style in the **Graph** panel, see “To specify a graph for a chart” on page 96.
- 8 To finish and save your chart, click **OK**.

## To create a trend chart

- 1 Select **Chart > Create**. The **Create Chart** dialog box.
- 2 From the **Chart Type** list, select **Trend**.
- 3 Provide information for the **General**, **Sharing**, and **Project Filter** tabs. For detailed information, see “Chart Dialog Box: General, Sharing, Query, and Project Filter Tabs” on page 104.
- 4 Click the **Charting Values** tab. The **Charting Values** panel displays.



Under **Time Values**, specify a date range for the chart data. Based on the **Start Date** and **End Date** values you select, additional lists and fields become available.

The following table contains possible **Start Date** and **End Date** values:

Value	Description
Date	<p>Specifies an exact date, selected from the calendar field. The default is the current system date.</p> <p>Specifying the date is useful for creating temporary charts with an exact date range.</p> <p><b>Note:</b> For the end date, selecting a date in the future uses the run date as the end date.</p>
Calculated Date	<p>Specifies a generic date range that is calculated by an interval type and number of intervals (see instructions proceeding this table).</p> <p>Use calculated dates for frequently used charts, such as charting the weekly defect count. You can set the generic date range once, eliminating the need to manually change it each week (or other interval).</p>
Run Chart Date (End Date only)	Specifies the chart's run date as the end date.

The following table contains the valid **Start Date** and **End Date** combinations:

Start Date	End Date
Date	Date
Date	Calculated Date
Date	Run Chart Date
Calculated Date	Date
Calculated Date	Run Chart Date

- From the **Interval Type** list, select the interval type to use for summarizing chart data list. Possible values are: Hours, Day, Week, Months, Quarter, and Year. For example, if you select Week, the line graph generated for the chart contains a point on the graph for each week.

When using a calculated date for either the start or end date, you need to specify a value for the interval to generate that calculated date.

- To specify a value for the interval, enter it in the **Number of interval type** field, where *interval type* is the interval type selected from the **Interval Type** list. For example, if you specify 5 with the **Interval Type** of Weeks, a line graph chart displays data during a five week period.

The chart's calculated end (or start) date is determined by the product of the interval type and the specified value. For example, setting a start date of 24-Apr-2010 with an interval of 5 weeks, generates an end date of 29-May-2010.

- Under **Values To Chart**, click the **Field** list and select the field you want to report on. For example, if you select **Type**, chart data is based on the value in the item **Type** field. The list of field values changes to display the available values for the field you select.
- Under **Values**, select the field value(s) to chart.

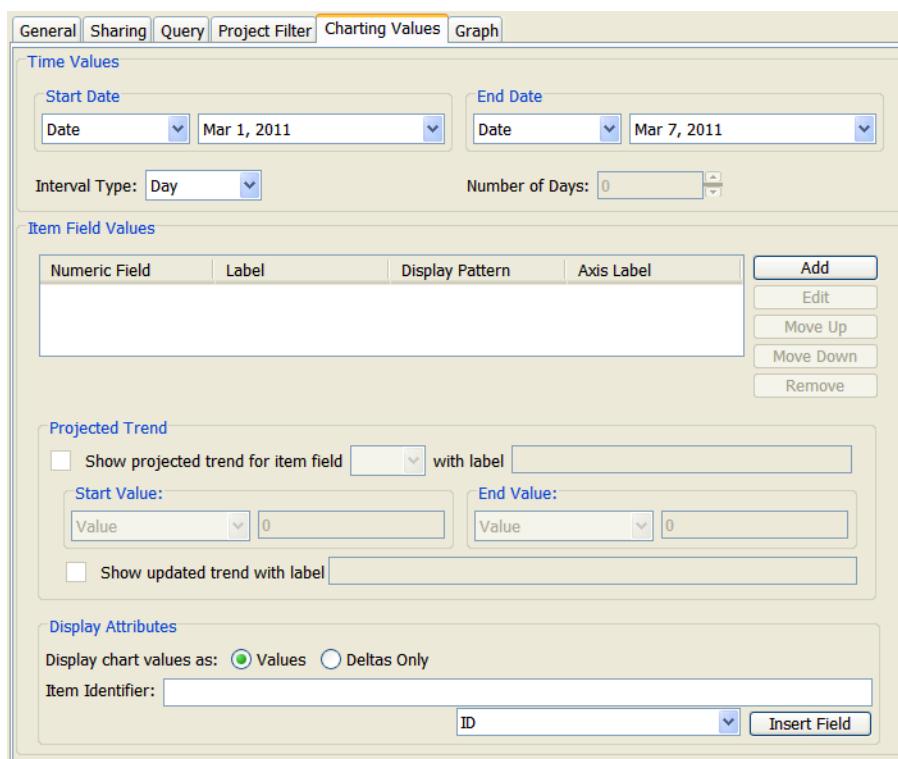
**NOTE** **Created By** and **Modified By** fields are not valid for trend charts and do not display in the list of fields.

- If you want the field value to display in the chart with a different name, select a field value in the **Values** list, then type or select the new name in the **Alias** field. If you want multiple field values to display on the chart combined under one name, select the field values in the Values list, then select or type a name in the **Alias** field. For example, if you are reporting on Docs, Feature Request, and Defect item types, and you assign an alias of Development to both Feature Request and Defect field values, the data for these items types is combined in the chart under Development.

- 10 To include in the chart automatically any new values that are defined for the field in the future, select **Include Future Values**.
- 11 To specify a graph style in the **Graph** panel, see “To specify a graph for a chart” on page 96.
- 12 To finish and save your chart, click **OK**.

### To create an item fields trend chart

- 1 Select **Chart > Create**. The **Create Chart** dialog box.
- 2 From the **Chart Type** list, select **Item Fields Trend**.
- 3 Provide information for the **General**, **Sharing**, **Query**, and **Project Filter** tabs. For detailed information, see “Chart Dialog Box: General, Sharing, Query, and Project Filter Tabs” on page 104.
- 4 Click the **Charting Values** tab. The **Charting Values** panel displays.



Under **Time Values**, specify a date range for the chart data. Based on the **Start Date** and **End Date** values you select, additional lists and fields become available.

The following table contains possible **Start Date** and **End Date** values:

Value	Description
Date	<p>Specifies an exact date, selected from the calendar field. The default is the current system date.</p> <p>Specifying the date is useful for creating temporary charts with an exact date range.</p> <p><b>Note:</b> For the end date, selecting a date in the future uses the run date as the end date.</p>
Item Date Field (Start Date only)	Specifies either Created Date (date item created) or Modified Date (date item last modified).
Calculated Date	<p>Specifies a generic date range that is calculated by an interval type and number of intervals (see instructions proceeding this table).</p> <p>Use calculated dates for frequently used charts, such as charting the weekly defect count. You can set the generic date range once, eliminating the need to manually change it each week (or other interval).</p>
Run Chart Date (End Date only)	Specifies the chart's run date as the end date.

The following table contains the valid **Start Date** and **End Date** combinations:

Start Date	End Date
Date	Date
Date	Calculated Date
Date	Run Chart Date
Item Date Field	Calculated Date
Calculated Date	Date
Calculated Date	Run Chart Date

- 5 From the **Interval Type** list, select the interval type to use for summarizing chart data list. Possible values are: Hours, Day, Week, Months, Quarter, and Year. For example, if you select Week, the line graph generated for the chart contains a point on the graph for each week.

When using a calculated date for either the start or end date, you need to specify a value for the interval to generate that calculated date.

- 6 To specify a value for the interval, enter it in the **Number of interval type** field, where *interval type* is the interval type selected from the **Interval Type** list. For example, if you specify 5 with the **Interval Type** of Weeks, a line graph chart displays data during a five week period.

The chart's calculated end (or start) date is determined by the product of the interval type and the specified value. For example, setting a start date of 24-Apr-2010 with an interval of 5 weeks, generates an end date of 29-May-2010.

- 7 To add a numeric field to report on, click **Add**. The **Add Expression** dialog box displays. For more information, see “Item Fields Trend Chart: Add/Edit Expression Dialog Box” on page 108.

To edit a numeric field, select it, and click **Edit**. The **Edit Expression** dialog box displays. For more information, see “Item Fields Trend Chart: Add/Edit Expression Dialog Box” on page 108.

If there are multiple numeric fields under **Item Field Values**, you can change the order of a numeric field by selecting it and clicking **Move Up** or **Move Down**.

To remove a numeric field from the chart, select it, and click **Remove**.

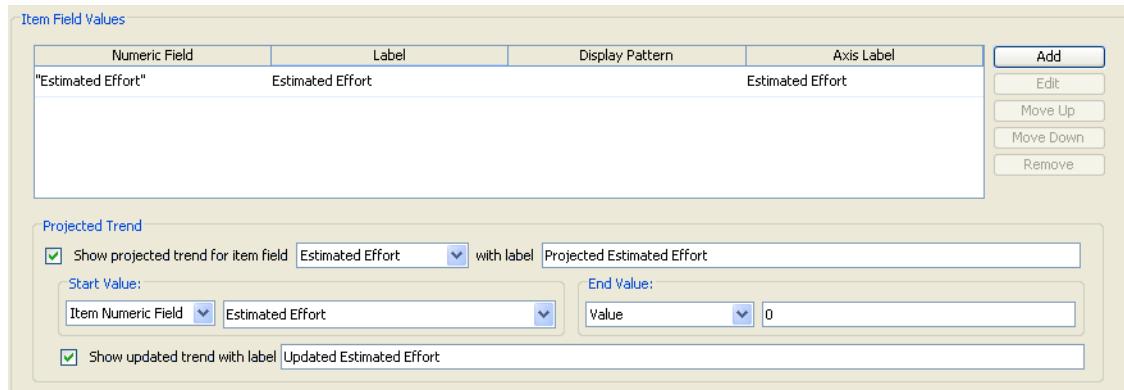
- 8 You can specify whether to display a projected trend for one of the displayed item field series, thus supporting the use of *burn-down charts* for planning work remaining within a fixed duration project. The projected trend displays a line between a starting value corresponding to the starting

date and the end value corresponding to the end date. In addition, if you choose to display the projected trend, you can also display the actual trend as a line between the last actual series value to the end value corresponding to the end date.

To include a projected trend graph in the chart, in the **Projected Trend** section, enable **Show projected trend for item field** and select a value from the list (populated by item field values added in the previous step) and, optionally, specify a label in the **with label** field. Next, specify a **Start Value** and **End Value**.

To display the projected trend, enable **Show updated trend** and, optionally, specify a label in the **with label** field.

For example:



**9** In the **Display Attributes** section, select from the following:

- **Display options:**
  - To display the current and previous values of the reported numeric fields, select the **Values** option.
  - To display the differences between the current and previous values of the reported numeric fields, select the **Deltas Only** option. Only item deltas between the chart start and end date are included in the chart.
  - In the **Item Identifier** field, specify the field you want to identify items by in the chart. For example, if you type {ID} each item in the chart is identified by the value of the **ID** field. An item identifier displays as an axis label in the chart. For better visibility, keep item identifiers short. You can manually insert fields into the **Item Identifier** field by selecting a field from the field list and then clicking **Insert Field**.

If you want to add text that precedes the field, type it before the field, for example, **Project: {ID}**. The chart then identifies each item by displaying **Project: ID field value**. You can specify multiple fields, for example, **Project: {ID} {State} {Priority}**.

**10** To specify a graph style in the **Graph** panel, see “To specify a graph for a chart” on page 96.

**11** To finish and save your chart, click **OK**.

### To specify a graph for a chart

- 1 In the **Create Chart** dialog box, click the **Graph** tab. The **Graph** panel displays.

- 
- 2** Select the **Graph Style** to use for the chart. The graph styles you can select vary depending on what type of chart you are creating and the chart options you specify. For detailed information on the available graph styles, see “Chart Dialog Box: Graph Tab” on page 105.

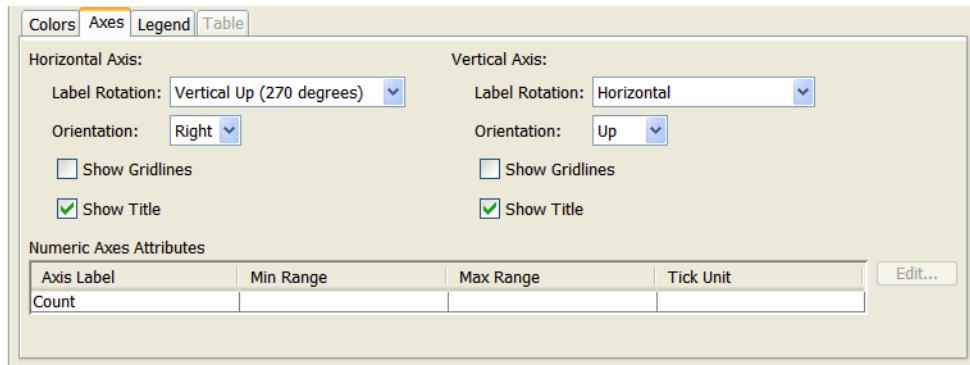
If you select the table graph style, different options become available in the **Graph** panel. For more information, see “To specify table graph style options” on page 97.

- 3** To change the default color used to outline the data on the chart, click **Edit** next to the **Data Outline Color** field. The **Select Color** dialog box displays. For detailed information, see “Select Color Dialog Box” on page 108.
- 4** Select a predefined color, or define a custom color for use in the chart.
- 5** To change the default color used for the chart background, click **Edit** next to the **Background Color** field. The **Select Color** dialog box displays.
- 6** To use custom colors for the chart data, select **Use Custom Data Colors** and click **Add**. The **Select Color** dialog box displays.
- 7** Click the **Axes** tab. The **Axes** panel displays.

---

**NOTE** If you choose a pie graph, the **Axes** tab is unavailable.

---



- 8** Horizontal and vertical axes are defined by the chart type or number of expressions included in the chart; however, you can edit the axes attributes. If you are creating a chart that uses numeric fields, you can also define numeric axes attributes.

---

**NOTE**

- If you specify one set of numeric axes attributes (minimum range, maximum range, and tick unit), these attributes are specified for the X and Y axes. For XY (scatter) charts, you should not set individual numeric axes attributes for the X and Y axes.
- For bubble charts, you should not specify numeric axes attributes because they override the calculated values provided by the underlying expression and users have to zoom in/out to properly view chart values.

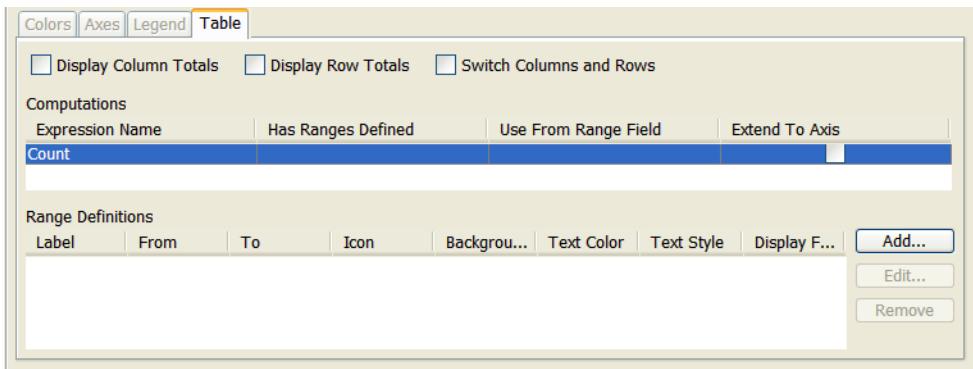
---

For detailed information on the **Axes** fields, see “Chart Dialog Box: Legend Tab” on page 106.

- 9** Click the **Legend** tab. The **Legend** panel displays.
- 10** Define the legend for your chart. For more information, see “Chart Dialog Box: Legend Tab” on page 106.

### To specify table graph style options

- 1** If you select the table graph style option in the **Graph** panel, the **Table** panel displays.



- 2 If your chart includes numeric field values and you want to display column totals in the table, select the **Display Column Totals** option. -Column Totals- displays under the **Expression Name** column.
- If your chart includes numeric field values and you want to display row totals in the table, select the **Display Row Totals** option. -Row Totals- displays under the **Expression Name** column.
- 3 To invert the display of columns and rows in your table, select the **Switch Columns and Rows** option.
- 4 If your table chart includes computed expressions, you can define ranges for the resulting numeric values. Computed expressions included in the table chart display in the **Computations** table. By default, the Count expression displays.

#### **NOTE**

- You cannot specify a range for the Count expression.
- You can specify a range for each computed expression; however, only one computed expression can specify the **Extend to Axis** option. The **Extend to Axis** option applies the background color of the selected computed expression to all computed expressions in the table chart.
- If a table cell contains a display definition that conflicts with the **Extend to Axis** option of another table cell, both table cells display the **Background Color** option of the table cell with the enabled **Extend to Axis** option.

- 5 To define a range for a computed expression, select the computed expression in the **Computations** table, and click **Add**. The **Add Range Display Definition** dialog box displays.
- 6 Use the **Label** field to assign a text string to the range category. Range labels can be up to 100 characters long.
- 7 In the **Value** field, use the **From** and **To** lists to specify values for the lower- and upper-range limits.

#### **NOTE**

- If a lower limit is not set, -Infinity is automatically entered in the **From** field. If an upper limit is not set, Infinity is automatically entered in the **To** field.
- A numeric value must be contained in one defined range; range intersections are invalid. For example, the following ranges are invalid: 0 – 5 and 4 – 8, or 0 – 5 and 5 – 10. For an integer field, an acceptable range is 0 – 5 and 6 – 10. For a floating point field, an acceptable range is 0 – 5 and 5.01 – 10.

- 8 To assign an icon image to the item, select **Use Custom Image**, and click **Select** to browse for the image file. To have no image associated with the type, select **No Image**. If you specify a custom icon image, the image must be in GIF or JPEG format, and no larger than 24 (width) by 16 (height) pixels.

- 
- 9 To change the default color used for the **Background Color** or **Text Color**, select **Custom** then click **Edit**. The **Select Color** dialog box displays.
  - 10 Select a pre-defined color, or define a custom color for the background color or text color.  
For detailed information on the **Select Color** tabs, see “Select Color Dialog Box” on page 108.
  - 11 To change the default text style, select a style from the **Text Style** list.
  - 12 To change the default display format, select a format from the **Display Format** list.
  - 13 To save the range display definition, click **OK**. The **Table** panel displays the new range display definition in the **Range Definitions** table.
  - 14 To edit a range for a computed expression, select the computed expression in the **Computations** table. The range displays in the **Range Definitions** table.  
Click **Edit**. The **Edit Range Display Definition** dialog box displays.  
To remove a range for a computed expression, select the computed expression in the **Computations** table. The range displays in the **Range Definitions** table.  
Click **Remove**. The range is removed from the **Range Definitions** table.
  - 15 To apply the range definition for a computed expression to all computed expressions in the chart, select a computed expression in the **Computations** table and enable the **Extend to Axis** option.
  - 16 To save the table chart, click **OK**.

## Detailed Chart Examples

This section provides details on defining the key components of sample charts of each type.

### Trend Chart

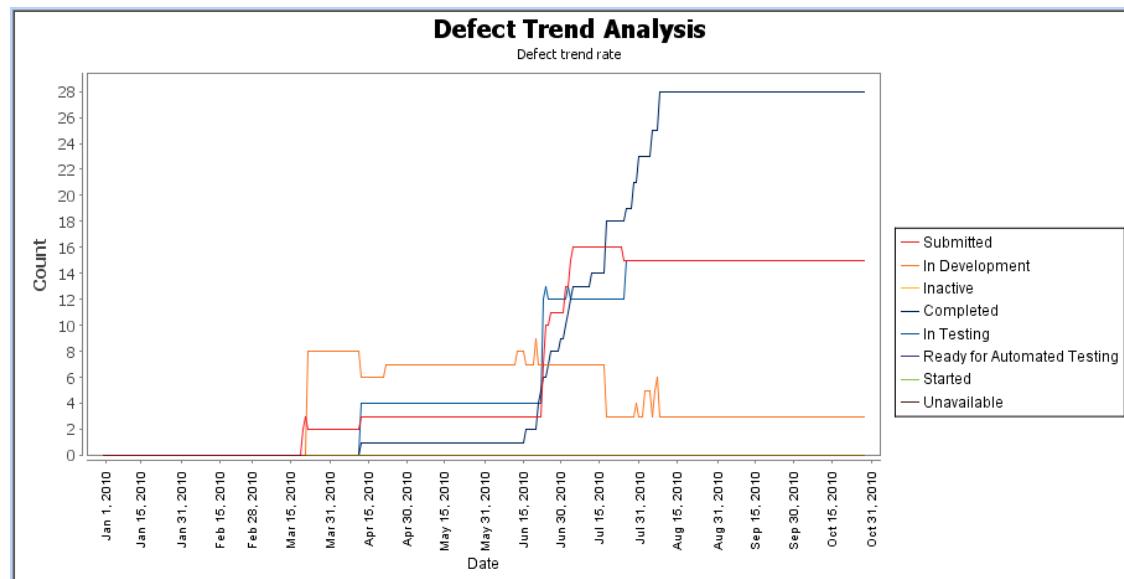
To create a trend chart that shows the number of defects created over the current year and their resolution (by state), summarized on a daily basis, do the following:

- 1 On the **Charting Values** panel, define the following values:

Field	Value
<b>Start Date</b>	Date, Jan 1, 2010
<b>End Date</b>	Run Chart Date
<b>Interval Type</b>	Day
<b>State</b>	State
<b>Values</b>	In Review [alias Submitted] Assigned [alias Submitted] In Development [alias In Development] Inactive Completed [alias Completed] In Testing [alias In Testing] Accepted [alias Submitted] Proposed [alias Submitted] Ready for Automated Testing Started Unavailable
<b>Include Future Values</b>	enabled

- 2 On the **Graph** panel, choose Line Graph as the **Graph Style**.

For example, the following chart is generated:



### Item Fields Trend Chart

To create an item fields trend chart that reports on the progression of test authoring activity over a period of weeks, do the following:

- 1 On the **Query** tab, select a query that shows the Project items that you want to chart.
- 2 On the **Charting Values** tab, under **Time Values**, define the following values:

Field	Value
<b>Start Date</b>	Item Date Field, Created Date
<b>End Date</b>	Calculated Date
<b>Interval Type</b>	Week
<b>Interval</b>	52

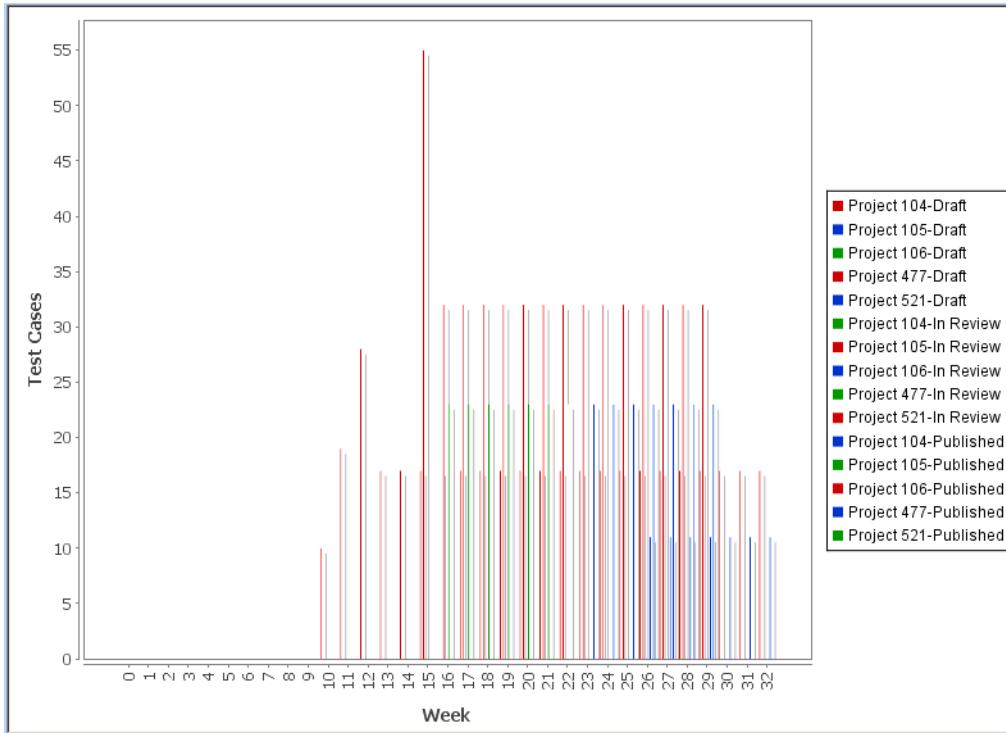
- 3 Under **Item Field Values**, define the following expressions:

Numeric Field	Expression Label
Test Case Content Count - Open	Draft
Test Case Content Count - In Review	In Review
Test Case Content Count - Published	Published

For all expressions, **Axis Label** is Test Cases and **Display Pattern** is #,###.

- 4 Under **Display Attributes**, select **Display chart values as: Values**.
- 5 On the **Graph** panel, choose Vertical Bar Graph as the **Graph Style**.
- 6 On the **Graph > Colors** panel, define custom colors for the Draft, In Review, and Published data.
- 7 Click **OK**.

For example, the following chart is generated:



### **Distribution Chart**

To create a distribution chart that shows the health and status of active projects per product, do the following:

- 1 On the **Query** tab, select a query that shows active Project items.
- 2 On the **Charting Values** tab, under **Values to Chart**, select **Product** as the **Field**, and then choose active products from the **Values** list.
- 3 Enable **Include Future Values** and **Show Zero Count Values**.
- 4 On the **Charting Values > Aggregation** panel, define the following expressions:

Aggregate Expression	Expression Label	Axis Label	Display Pattern
count()	# of Projects	Count	#,###
max("Product Health Value")	Health	Count	none
sum("Total Estimated Effort")	Total Estimated Effort	Days	,##0 days
sum("Total Actual Effort")	Total Actual Effort	Days	,##0 days
sum("Total Estimated Effort") - sum("Total Actual Effort")	Effort Variance	Days	,##0 days

- 5 On the **Graph** panel, choose Table Rendering as the **Graph Style**.
- 6 On the **Graph > Table** panel, define the following **Range Definitions** for the **Health** computed expression:

Label	From	To	Icon	Text Style	Display Format
Green	-1.0	-1.0	green circle image	Plain	Icon
Yellow	0.0	0.0	yellow circle image	Plain	Icon
Red	1.0	1.0	red circle image	Plain	Icon

Define the following **Range Definitions** for the Effort Variance computed expression:

Label	From	To	Icon	Text Style	Display Format
Red	-Infinity	0.0	red circle image	Plain	Icon & Value
Yellow	1.0	10.0	yellow circle image	Plain	Icon & Value
Green	11.0	Infinity	green circle image	Plain	Icon & Value

## 7 Click OK.

For example, the following chart is generated:

Product	# of Projects	Health	Total Estimated Effort	Total Actual Effort	Effort Variance
Adaptive Cruise Control System - (103)	1	🟡	27,500 days	0 days	🟡 27,500 days
Cardiac Device - (102)	0				
Integrated Patient Monitor and Display - (101)	1	🟡	7,250 days	0 days	🟡 7,250 days
Tempus Classic - (98)	3	🟡	11,850 days	0 days	🟡 11,850 days
Tempus Modern - (100)	0				
Tempus Rugged - (99)	0				

## Item Fields Charts

To create an item fields chart that shows effort, budget, and schedule metrics for all active projects, do the following:

- 1 On the **Queries** tab, select a query that shows active Project items.
- 2 On the **Charting Values** tab, define the following expressions:

Field Name/Expression	Expression Label	Display Pattern
"Health Value"	Health	
"Estimated Budget"	Est. Budget	\$#,###
"Actual Budget"	Act. Budget	\$#,###
"Budget Variance"	Budget Variance	\$#,###
"Estimated Effort"	Est. Effort	#,###.## hrs
"Actual Effort"	Act. Effort	#,###.## hrs
"Effort Variance"	Effort Variance	#,###.## hrs

- 3 In the **Item Identifier** field, specify `{Project} ({Product})`.
- 4 On the **Graph > Table** panel, define the following **Range Definitions** for the `Health` computed expression:

Label	From	To	Icon	Text Style	Display Format
Green	-1.0	-1.0	green circle image	Plain	Icon
Yellow	0.0	0.0	yellow circle image	Plain	Icon
Red	1.0	1.0	red circle image	Plain	Icon

Define the following **Range Definitions** for the Budget Variance computed expression:

Label	From	To	Icon	Text Style	Display Format
Red	-9.9999999E8	-100001.0	red circle image	Plain	Icon & Value
Yellow	-10000.0	10.0	yellow circle image	Plain	Icon & Value
Green	10000.0	9.9999999E8	green circle image	Plain	Icon & Value

Define the following **Range Definitions** for the Effort Variance computed expression:

Label	From	To	Icon	Text Style	Display Format
Red	0.1500000000000001	9999.0	red circle image	Plain	Icon & Value
Yellow	1.0E-13	0.15	yellow circle image	Plain	Icon & Value
Green	-9999.0	0.0	green circle image	Plain	Icon & Value

5 Click **OK**.

For example, the following chart is generated:

Project Details							
Items	Health	Est. Budget	Act. Budget	Budget Variance	Est. Effort	Act. Effort	Effort Variance
/Embedded/Projects/Release1 (Tempus Classic - (98))	●	\$350,000	\$375,000	● -\$25,000	4,900 hrs	0 hrs	● -1 hrs
/Medical/Projects/Release1 (Integrated Patient Monitor and Display - (101))	●	\$1,200,000	\$1,375,000	● -\$175,000	7,250 hrs	0 hrs	● -1 hrs
/Automotive/Projects/Release1 (Adaptive Cruise Control System - (103))	●	\$3,300,750	\$3,154,000		27,500 hrs	0 hrs	● -1 hrs
/Embedded/Projects/Release2 (Tempus Classic - (98))	●	\$490,000	\$600,000	● -\$110,000	5,750 hrs	0 hrs	● -1 hrs
/Embedded/Projects/Hardware (Tempus Classic - (98))	●	\$205,000	\$215,000		1,200 hrs	0 hrs	● -1 hrs

## Chart Dialog Box: General, Sharing, Query, and Project Filter Tabs

Tab	Field	Description
General	Name	<p>Type chart name as you want it to display in <b>Charts</b> menu. Chart name is mandatory. You cannot have two charts with same name.</p> <p>Next to the <b>Name</b> field,  displays, indicating that the chart is a <i>favorite</i>. Favorites are MKS Integrity objects (queries, charts, reports, and dashboards) that you created and use.</p>
	Is Admin Provided	<p>To convert the chart to an admin provided object, enable <b>Is Admin Provided</b>. Admin provided objects are objects that are used by your administrator to define the MKS Integrity solution template for all users. For more information, see your administrator.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>■ This option is available only if the <code>createSharedAdmin</code> or <code>Admin</code> permission is assigned to you.</li> <li>■ Once user object is converted to admin provided object, you cannot revert to user object again.</li> <li>■ Selecting <b>Is Admin Provided</b> option automatically adds currently logged in user to <b>Sharing</b> tab with edit permissions.</li> </ul>
	Title	<p>Type chart title as you want it to display on chart. Title is optional. To choose font for title, click <b>Font</b>, then select font and size from <b>Select Font</b> dialog box. If no font selected, default font is assigned.</p>
	Description	<p>Type chart description. Description is optional.</p>
	Show Description	<p>Select to show description on chart. To choose font for description, click <b>Font</b>, then select font and size from <b>Select Font</b> dialog box. If no font selected, default font is assigned.</p>
	Footnote	<p>Type chart footnote. Footnote is optional. To choose font for footnote, click <b>Font</b>, then select font and size from <b>Select Font</b> dialog box. If no font selected, default font is assigned.</p>
Sharing		<p>To select the principals (users and groups) you want to share your chart with, click <b>Choose Principals</b> and use the data filter.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>■ The list of available groups is limited to those with the same project permissions as the group you are in.</li> <li>■ You cannot share a chart to the <code>everyone</code> group unless you are an administrator or you have the <code>ShareToEveryone</code> permission. Contact your administrator for more information.</li> </ul> <p>To allow assigned principals to edit the chart, select the principal's check box under the <b>Modify</b> column. To deny assigned principals the ability to edit the chart, clear the principal's check box under the <b>Modify</b> column. By default, all assigned principals are denied the ability to edit a chart.</p>
Query		<p>Click the <b>Query</b> list and select a query to base the chart on. Any filters used in the query apply to the chart.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>■ You should base charts on queries that do not have defined project filters. This enables you to re-use the same chart for different projects by applying the project filter at run time.</li> <li>■ You cannot select a query for a trend chart.</li> </ul>
Project Filter		<p>Select the project(s) you want to filter the chart data on when it is run.</p> <p><b>Note:</b> You can also define project filters for dashboards. Depending on how you design your dashboard, when you run a chart in a dashboard, the dashboard's project filter may override the chart's project filter.</p>

## Chart Dialog Box: Graph Tab

Field	Description
<b>Graph Style</b>	<p>Line graphs display trends over time or categories. Available for trend and item fields trend charts only.</p> <hr/> <p>Vertical bar graphs compare field values across categories. Available for all chart types.</p> <hr/> <p>Vertical stacked bar graphs display relationship of single items to whole. Available for all chart types.</p> <hr/> <p>Horizontal bar graphs compare field values across categories. Available for distribution and item fields charts only.</p> <hr/> <p>Horizontal stacked bar graphs display relationship of single items to whole. Available for distribution and item fields charts only.</p> <hr/> <p>Pie graphs display contribution of each field value to total. Available for distribution and item fields charts only. Distribution charts cannot contain grouping of field values and multiple aggregate expressions.</p> <hr/> <p>XY (scatter) graphs display relationships among numeric field values in groups of field data. Available only for distribution and item fields charts that include two aggregate expressions and no grouping of field values.</p> <hr/> <p>Bubble graphs are similar to XY (scatter) graphs; however, they compare three groups of field data. Size of bubble indicates value of third group of field data. Available only for distribution and item fields charts that include three aggregate expressions and no grouping of field values.</p> <hr/> <p>Table graphs display field data in simple table. Available for all chart types.</p>
<b>3D Chart</b>	Click to create three dimensional chart.
<b>Display Shapes (Line Graphs)</b>	If you select line graph and you want to display shapes in chart, select this option. Shapes in chart represent data allowing you to differentiate data in chart more easily.
<b>Display Labels</b>	Click to display labels for values in chart. If you select pie graph style, this option is automatically selected.

## Chart Dialog Box: Axes Tab

Field	Description
<b>Horizontal Axis: Label Rotation</b>	<p><b>Horizontal</b> displays label text horizontally.</p> <p><b>Vertical Down (90 degrees)</b> displays label text vertically at 90 degrees from horizontal axis.</p> <p><b>Vertical Up (270 degrees)</b> displays label text vertically at 270 degrees ending at horizontal axis.</p> <p><b>45 Down (45 degrees)</b> displays label text vertically at 45 degrees from horizontal axis.</p> <p><b>45 Up (315 degrees)</b> displays label text vertically at 45 degrees ending at horizontal axis.</p>
<b>Horizontal Axis: Orientation</b>	Chart values on horizontal axis display starting from right or left.
<b>Horizontal Axis: Show Gridlines</b>	Grid lines on horizontal axis.
<b>Horizontal Axis: Show Title</b>	Title for horizontal axis.
<b>Vertical Axis: Label Rotation</b>	<p><b>Horizontal</b> displays label text horizontally.</p> <p><b>Vertical Up (270 degrees)</b> displays label text vertically at 270 degrees from bottom of vertical axis.</p>
<b>Vertical Axis: Orientation</b>	Chart values on vertical axis up or down from horizontal axis.

Field	Description
<b>Vertical Axis:</b> <b>Show Gridlines</b>	Grid lines on vertical axis.
<b>Vertical Axis:</b> <b>Show Title</b>	Title for vertical axis.
<b>Numeric Axes Attributes:</b> <b>Axis Label</b>	Name for numeric axis as it displays in chart. Displays only for charts that use numeric field values and is specified when you create underlying aggregate or computed expression. If you do not specify axis name, default name is specified.
<b>Numeric Axes Attributes:</b> <b>Min Range</b>	For charts that use numeric field values, click <b>Edit</b> . In <b>Min Range</b> field, type minimum range to display numeric field values in chart. If you do not specify range, default range displays in chart.
<b>Numeric Axes Attributes:</b> <b>Max Range</b>	For charts that use numeric field values, click <b>Edit</b> . In <b>Max Range</b> field, type maximum range to display numeric field values in chart. If you do not specify range, default range displays in chart.
<b>Numeric Axes Attributes:</b> <b>Tick Unit</b>	For charts that use numeric field values, click <b>Edit</b> . In <b>Tick Unit</b> field, specify units that display on numeric axis. For example, if you specify minimum range of 0, maximum range of 100, and tick unit of 10, numeric axis displays 0, 10, 20, 30, 40, and so on up to 100.

## Chart Dialog Box: Legend Tab

Field	Description
<b>Show Legend</b>	Show legend on chart.
<b>Title</b>	Title for chart legend. Title is optional.
<b>Position</b>	Position for legend on chart.
<b>Background Color</b>	Click <b>Edit</b> , and then select color from <b>Select Color</b> dialog box.

## Distribution Chart: Add/Edit Expression Dialog Box

The **Add/Edit Expression** dialog box allows you to create or edit an aggregate expression used in a distribution chart.

- In the **Aggregation Expression** field, create an aggregate expression that incorporates a numeric field in the items returned by the specified query. For example, if your chart currently displays Project items by state, typing: `sum("Actual Budget")` adds the **Budget** fields in each state grouping of Project items to display a total budget by state. You can also manually insert fields into your expression by selecting a field from the field list and clicking **Insert Field**.

Depending on the number of aggregate expressions you create and whether you choose to group field values, only certain graph styles are available. For example, if you do not group field values and you include two aggregate expressions, the XY (scatter) graph style is available. If you do not group field values and you include three aggregate expressions, the bubble style graph is available. For more information on selecting graph styles, see “To specify a graph for a chart” on page 96.

- In the **Expression Label** field, type a name for the computed expression label as you want it to display in the chart. For example, if you have an aggregate expression that adds the **Budget** fields in each state grouping of Project items to display a total budget by state, an appropriate label might be **Total Budget**. If you do not define a label, the computed expression displays, for example, `sum("Actual Budget")`.

- 
- 3 In the **Display Pattern** field, select or type a display pattern for the value of the computed expression.
  - 4 In the **Axis Label** field, do one of the following:
    - If your chart contains one aggregate expression, type a name for the numeric axis as you want it to display in the chart.
    - If your chart contains multiple aggregate expressions and you want to specify the same axis label for all aggregate expressions, type a name for the numeric axis as you want it to display in the chart, or select an existing axis label from list.
    - If your chart contains multiple aggregate expressions and you want to specify an axis label for each aggregate expression, type a name for the numeric axis as you want it to display in the chart.
  - 5 To add the aggregate expression to the chart, click **OK**.

## Item Fields Chart: Add/Edit Expression Dialog Box

The **Add/Edit Expression** dialog box allows you to create or edit a computed expression for an item fields chart.

- 1 In the **Field Name/Expression** field, create a computed expression that incorporates one or more numeric fields in the items returned by the specified query. For example, if your chart currently displays a list of Project items, typing: ("Actual Budget" - "Estimated Budget") calculates the difference between the **Actual Budget** and **Estimated Budget** fields for each Project item and displays the values in the chart. You can also manually insert fields into your expression by selecting a field from the field list and clicking **Insert Field**. For more information on creating computed expressions, see "What are Computed Expressions?" on page 159.

Depending on the number of computed expressions you create, certain graph styles become available. For example, if you include two computed expressions, the XY (scatter) graph style is available. If you include three computed expressions, the bubble style graph is available. For more information on selecting graph styles, see "To specify a graph for a chart" on page 96.
- 2 In the **Expression Label** field, type a name for the computed expression label as you want it to display in the chart. For example, if you have a computed expression that calculates the difference between the **Actual Budget** and **Estimated Budget** fields for each Project item, an appropriate label might be Over/Under Budget. If you do not define a label, the computed expression displays, for example, ("Actual Budget" - "Estimated Budget").
- 3 In the **Display Pattern** field, select or type a display pattern for the value of the computed expression.
- 4 In the **Axis Label** field, do one of the following:
  - If your chart contains one computed expression, type a name for the numeric axis as you want it to display in the chart.
  - If your chart contains multiple computed expressions and you want to specify the same axis label for all computed expressions, type a name for the numeric axis as you want it to display in the chart, or select an existing axis label from list.
  - If your chart contains multiple computed expressions and you want to specify an axis label for each computed expression, type a name for the numeric axis as you want it to display in the chart.
- 5 To add the computed expression to the chart, click **OK**.

---

## Item Fields Trend Chart: Add/Edit Expression Dialog Box

The **Add/Edit Expression** dialog box allows you to create or edit the details of a numeric field reported on in an item fields trend chart.

- 1 From the **Numeric Field** list, select the numeric field you want to report on.
- 2 In the **Expression Label** field, type a name for the numeric field as you want it to display in the chart.
- 3 In the **Display Pattern** field, select or type a display pattern for the value of the numeric field.
- 4 In the **Axis Label** field, do one of the following:
  - If your chart contains one numeric field, type a name for the numeric axis as you want it to display in the chart.
  - If your chart contains multiple numeric fields and you want to specify the same axis label for all numeric fields, type a name for the numeric axis as you want it to display in the chart, or select an existing axis label from list.
  - If your chart contains multiple numeric fields and you want to specify an axis label for each numeric field, type a name for the numeric axis as you want it to display in the chart.
- 5 To add the numeric field to the chart, click **OK**.

## Select Color Dialog Box

Tab	Description
<b>Swatches</b>	Select pre-defined color from palette.
<b>HSB (Hue, Saturation, Brightness)</b>	Modify color's hue, saturation, and brightness. To increase or decrease elements, drag color slider or enter numeric color values.
<b>RGB (Red, Green, Blue)</b>	Modify red, green, and blue colors. To increase or decrease colors, drag color slider or enter numeric color values.

---

# Editing a Chart

---

**CLI EQUIVALENT** im editchart

---

In the GUI, you can edit the details of a chart; however, you cannot change the chart type. Changes to chart criteria affect the data displayed the next time the chart is run.

---

**NOTE** You cannot edit charts in the Web interface.

---

To edit a chart, select a chart from the **Manage Charts** view, then select **Chart > Edit**.

## **Key Considerations**

- If the chart you are editing is based on a query that is invisible to you, the query does display in the **Query** list. However, this query is not available when you are creating a chart.
- You can edit the charts you have created. Principals (users and groups) that a chart is shared with can edit it if they have edit permissions assigned to them by the chart creator.
- All charts are subject to visibility rules set by your administrator. Visibility rules restrict access to specific information based on project and/or item type. For more information, see the *MKS Integrity Server 2009 Administration Guide*, or see your administrator.
- In addition to the tabs specific to the chart type, the following tabs display in the **Chart Properties** dialog box:
  - **References** displays all admin provided and user objects that reference the chart. If you plan on making changes to the chart or deleting it, this information lets you know what objects are affected by the changes or deletion.  
For more information on admin provided objects, see the *MKS Integrity Server 2009 Administration Guide*.
  - **History** displays all changes made to the chart.

# Running a Chart

**CLI EQUIVALENT** im runchart

In the GUI and Web interface, you can run charts. You can also run charts through the dashboard.

While viewing a distribution chart, item fields chart, or item fields trend chart, you can select any chart value and drill down to see the specific items or details of an item that make up that value.

## Key Considerations

- All charts are subject to visibility rules set by your administrator. Visibility rules restrict access to specific information based on project and/or item type. For more information, see the *MKS Integrity Server 2009 Administration Guide*, or see your administrator.
- For the best performance, avoid generating trend or item fields trend charts using short intervals over long time spans.
- Displayed date fields do not change based on the time zone a user operates in; however, displayed date/time fields and time entries vary based on the time zone a user operates in.

## To run a chart in the GUI

- 1 From the **Manage Charts** view, select a chart, then select **Chart > Run**.

The chart displays in the **Chart** view.

Project Details							
Items	Health	Est. Budget	Act. Budget	Budget Variance	Est. Effort	Act. Effort	Effort Variance
/Projects/Release1 (Tempus Classic - (98))	●	\$350,000	\$375,000	4,900 hrs	0 hrs	● -1 hrs	
/Projects/Release2 (Tempus Classic - (98))	●	\$490,000	\$600,000	5,750 hrs	0 hrs	● -1 hrs	
/Projects/Hardware (Tempus Classic - (98))	●	\$205,000	\$215,000	1,200 hrs	0 hrs	● -1 hrs	

An item fields chart using a table graph style to display Effort, Budget, and Schedule metrics for all active projects.

With the exception of table charts, you can resize all charts by right-clicking and selecting **Zoom In** or **Zoom Out**. For bar charts, you also need to select the axis to resize. The axes you can select depend on the graph style of the chart. Select **Auto Range** to restore the chart to its original size.

You can sort a table chart by column or row. Click a column to sort the rows in ascending or descending order for the values in that column; click a row to sort the columns in ascending or descending order for the values in that row.

- 2 For distribution charts, if you want to view the items that make up a particular chart value, click the chart value (either an area of a bar, pie, or trend style chart, or a cell in a table style chart). The **Items** view displays only the items that make up that portion of the chart's data set.

The items are filtered based on the project filter used to produce the chart results. You cannot change the project filter.

**NOTE** If you have set up a project filter preference for the **Items** view, it is disregarded until you run another query.

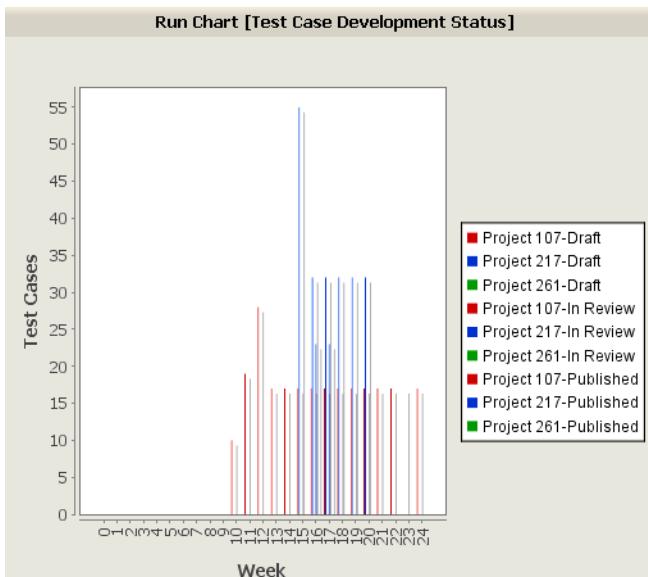
With the exception of trend charts, you can view a single item that makes up a particular chart value by clicking the chart value (either an area of a bar, pie, or a cell in a table style chart). The

---

**Item Detail** view displays detailed information about the item that makes up that portion of the chart's data set.

### To run a chart in the Web interface

- 1 In the **Charts View**, select the chart you want to run, and select **Run Chart**. The chart displays in the **Chart** view.



*An item fields trend chart using a vertical bar graph style to display development status of all test cases.*

- 2 With the exception of table and pie charts, you can click **Zoom Out** to zoom out on the axes or **Zoom In** to zoom in on the axes.
- 3 To increase the size of the chart, click **Increase Size**, and to decrease the size of the chart, click **Decrease Size**.
- 4 To refresh the contents of the chart, click **Refresh View**. You can sort a table chart by column or row. Click a column to sort the rows in ascending or descending order for the values in that column; click a row to sort the columns in ascending or descending order for the values in that row.
- 5 For distribution charts, if you want to view the items that make up a particular chart value, click the chart value (either an area of a bar or pie chart, or a cell in a table chart). The **Items View** displays only the items that make up that portion of the chart's data set.

The items are filtered based on the project filter used to produce the chart results. You cannot change the project filter.

For item fields and item fields trend charts, if you want to view a single item that makes up a particular chart value, click the chart value (either an area of a bar or pie chart, a cell in a table chart, or a point on a line graph chart). The **Item Detail** view displays detailed information about the item that makes up that portion of the chart's data set.

---

# Copying a Chart

---

**CLI EQUIVALENT** `im copychart`

---

In the GUI, you can copy the details of a chart and save them as a new chart with a different name.

---

**NOTE** You cannot copy charts in the Web interface.

---

To copy a chart, select a chart from the **Manage Charts** view, then select **Chart > Copy**.

## **Key Considerations**

- A copied favorite ( or non-favorite () chart becomes a favorite when you save the new chart.
- If the chart you are copying is an admin provided object, the **Is Admin Provided** option is cleared in the copy.
- Sharing information is not copied. You can copy sharing information by clicking **Copy Principals** on the **Sharing** tab.

---

**NOTE** You cannot share a chart to the `everyone` group unless you are an administrator or you have the `ShareToEveryone` permission. Contact your administrator for more information.

---

---

# Saving and Printing a Chart

When you run a chart, you can save the chart in the **Chart** view as an image file or print the chart.

---

**NOTE** In the GUI, you can only save the chart as a PNG image. In the Web interface, you can only save the chart as a PNG or BMP image.

---

The following tables describe how to save and print a chart:

Operation	Interface	Procedure
Saving a chart as an image	GUI	Click <b>Save As</b> .
	Web	Right-click the chart and, depending on your default browser, select either <b>Save Picture As</b> or <b>Save Image As</b> .
Printing a chart	GUI	Click <b>Print</b> .
	Web	Select <b>Print Chart</b> .

---

# Deleting a Chart

---

**CLI EQUIVALENT** `im deletechart`

---

In the GUI, you can delete charts you no longer use. You can only delete charts in the GUI, and you can only delete charts you created, unless you have administrator permissions.

The following table describes how to delete a chart:

Interface	Procedure
GUI	From the <b>Manage Charts</b> view, select a chart, then select <b>Chart &gt; Delete</b> .
Web	From the <b>Charts View</b> , select a chart, and select <b>Delete Chart</b> .

## PART 4

# Viewing Item Data in Reports

---

## What Is a Report?

A *report* is a summary of data. Reports are based on existing queries. The filters used in the query apply to the report. You decide the layout format to use for the report, which fields to include in the report, and the order the fields appear in. You can also group and sort the report data.

---

# Reports View: Managing Reports

---

**CLI EQUIVALENT** `im reports`

---

You can view and manage all reports created by you or shared to you from the **Manage Reports** view (GUI) or **Reports View** (Web interface).

To display the **Manage Reports** view in the GUI, select **Report > Manage Reports**.

To display the **Reports View** in the Web interface, under **Quick Views**, click **Reports**.

The data filter in this view displays your favorite reports (reports that you created and use) by default. You can search for a specific report by typing in the text filter and/or specifying a filter.

In this view, you can:

- run a report (see “Running a Report” on page 131)
- create a report (see “Creating a Report” on page 119)
- copy a report (see “Copying a Report” on page 130)
- delete a report (see “Deleting a Report” on page 133)
- edit a report (see “Editing a Report” on page 129)
- view a report definition (see “Report Definition View” on page 118)

---

# Report Definition View

This view displays the details of a report definition.

To view specific information about the report, click a tab. You cannot edit the report definition while viewing it.

---

**NOTE** If the report you are viewing is based on a query invisible to you, the query does display in the **Query** field; however, this query is not available when you are creating a report.

---

# Creating a Report

---

**CLI EQUIVALENT** im createreport

---

The **Report Wizard** guides you through the steps required to create a report. The wizard works the same way in the GUI and in the Web interface. When creating a report, you select a report type that determines how the report data is organized.

MKS Integrity provides default report types; however, your administrator can create custom report types. The two report types are:

- **Basic reports**

Basic reports allow you to select the item fields to include in the report and specify how the report data sorts.

- **Detail reports**

Detail reports allow you to select component fields (item fields, change package fields, attachment fields, time entry fields, and fields on related items) to include in the report and how the report data sorts.

## **Key Considerations**

- The report types, template preview images, styles, and logos available in the wizard are MKS defaults and/or created by your administrator. To modify these elements or create your own, see your administrator or the *MKS Integrity Server 2009 Administration Guide*.
- Only HTML based report types display long text fields formatted with rich content and allow HTML formatting in the output. For XML and CSV report types, HTML formatting options do not display in the report wizard panels and any HTML tags in rich content fields are discarded.
- Reports can do more than just display field information. You can also perform arithmetic calculations between numeric fields, displaying the values in the report. For example, you can add up column totals or count the number of items in a specific state. To perform these calculations, you create a *computed expression*.
- You cannot create or edit a query while creating a report.
- A report can be edited by the user who created it. Principals (users and groups) that a report is shared with can edit it if they have edit permissions assigned to them by the report creator. Only the report creator or an administrator can delete the report.
- Because reports are based on queries, reports are subject to visibility rules set by your administrator. Visibility rules restrict access to specific information based on project and/or item type. For more information, see the *MKS Integrity Server 2009 Administration Guide*, or contact your administrator.
- Symbolic dates in rules and queries are evaluated on the MKS Integrity Client's time zone.
- If you do not specify a date or date/time format where applicable, the following defaults are provided: date format with a standard locale-specific date or date/time format with a standard locale-specific date that includes hours and minutes.
- Relevance and editability rules are evaluated on the MKS Integrity Client's time zone.
- Computed expressions return dates/times in the MKS Integrity Client's time zone and perform calculations in the MKS Integrity Server's time zone where appropriate.

- 
- Report types “Detail - HTML, Column, Relationships Context” and “Detail - HTML, Column, Relationships Structure” result in large reports that can take a long time to display in a browser window.
  - The report type “Basic - HTML, Schedule” displays one month before and after the current day. Values that exceed 30 days are ignored.
  - Creating deeply nested reports with a large number of inter-related items can create extremely large reports and/or cause the MKS Integrity Server to stop responding. When creating a report, take into consideration that the average number of links per item and the number of levels in the report multiply the size of the report.
  - Although the electronic signature fields **Signed By** and **Signature Comment** are only visible in an item’s history (if enabled by your administrator), you can report on the historical values by specifying the fields in the report wizard.

### To create a report

1 Do one of the following:

- In the GUI, select **Report > Create**.
- In the Web, from the **Reports View**, select **Create Report**.

The **Report Wizard** displays the **Type** panel.

2 From the list, select a report type for your report. Each report type displays a description and preview.

---

**NOTE** Depending on the report type you select, the remaining steps in this procedure may or may not apply. In addition, the order of the steps may be slightly different.

---

3 Provide information for the **Attributes**, **Style**, **Logo**, and **Parameters** panels. For more information, see “Report Wizard” on page 127.

4 Click **Next**. The **Item Fields** panel displays.

5 Select which fields you want to include in your report:

- specifies fields to add to the **Fields in Report** list using the data filter.
- removes selected fields from the **Fields in Report** list.

6 To specify the format layout of a field as it displays in the report, select a field in the **Selected** column and click . The **Format layout for <field name>** dialog box displays. For more information, see “Format Layout for field name Dialog Box” on page 126. Formatted fields are annotated with **(Formatted)** after the field name.

7 If your report includes numeric fields, you can perform arithmetic calculations between one or more numeric fields and display the results in the report. For example, if your report currently displays a list of Project items, you could calculate the difference between the **Actual Budget** and

---

**Estimated Budget** fields for each Project item and display the values in a new field called **Over/Under Budget**. The arithmetic calculation behind the **Over/Under Budget** is a computed expression.

To create a computed expression, click **Add** (GUI) or  (Web). The **Add Expression** dialog box displays. For more information, see “Add/Edit Expression Dialog Box” on page 126.

---

**NOTE** To create a valid computed expression, the numeric fields used in the expression must be in the **Selected** list of the previous panel.

---

To edit an existing computed expression, select it and click **Edit** in the GUI, or  in the Web. The **Edit Expression** dialog box displays. For more information, see “Add/Edit Expression Dialog Box” on page 126.

To remove a computed expression, select it, and click **Remove** (GUI) or  (Web).

- 8 Click **Next**. Depending on the report type you selected in the **Type** panel, one or more of the following panels may display in varying order or as segment sub-nodes:

- “Report Wizard: Segment Attributes Panel” on page 121
- “Report Wizard: Group By Panel” on page 122
- “Report Wizard: Field Computations Panel” on page 122
- “Report Wizard: Change Package Type Panel” on page 123
- “Report Wizard: Change Package Attributes Panel” on page 123
- “Report Wizard: Change Package Entry Attributes Panel” on page 123
- “Report Wizard: Attachment Fields Panel” on page 123
- “Report Wizard: Test Result Fields Panel” on page 124
- “Report Wizard: Test Step Results Panel” on page 124
- “Report Wizard: Test Result Attachment Attributes Panel” on page 124
- “Report Wizard: Test Result Related Items Panel” on page 125
- “Report Wizard: Time Entry Fields Panel” on page 125
- “Report Wizard: Relationship Fields Panel” on page 125
- “Report Wizard: Relationship Filter Panel” on page 125

- 9 Complete the options for each panel that displays. Where applicable, click **Next** to advance to the next panel. The **Sort By** panel displays.

- 10 Provide information for the **Sort By** panel. For more information, see “Report Wizard: Sort By Panel” on page 122.

- 11 Click **Next**. The **Sharing Panel** displays.

- 12 Provide information for the **Sharing** panel. For more information, see “Report Wizard” on page 127.

- 13 To save the report, click **Finish**.

## Report Wizard: Segment Attributes Panel

A *segment* is a subsection of item information defined in a report, for example, time entry or relationship field information. Segments are defined in the report type you select.

---

You can choose which segments to include in the report by selecting the segment node and enabling or clearing the **Enable segment** option in the segment panel. Disabled segments are indicated by  and disabled segment sub-nodes display in italics. A description of the selected segment displays in the segment panel.

## Report Wizard: Group By Panel

The **Group By** panel specifies how results are grouped together in a report. Depending on the report type you selected, the **Group By** panel may contain multiple **Group by level** tabs.

For each tab, select a field from the **Field** list, and then select the sort order (**Ascending** or **Descending**).

If your report includes numeric fields, you can perform arithmetic calculations between one or more numeric fields and display the results in the report. For example, if your report currently displays a list of Project items, you could calculate the difference between the **Actual Budget** and **Estimated Budget** fields for each Project item and display the values in a new field called **Over/Under Budget**. The arithmetic calculation behind the **Over/Under Budget** is a computed expression.

To create a computed expression, click **Add** (GUI) or  (Web). The **Add Expression** dialog box displays. For more information, see “Add/Edit Expression Dialog Box” on page 126.

---

**NOTE** To create a valid computed expression, the numeric fields used in the expression must be in the **Selected** list of the previous panel.

---

To edit an existing computed expression, select it and click **Edit** in the GUI, or  in the Web. The **Edit Expression** dialog box displays. For more information, see “Add/Edit Expression Dialog Box” on page 126.

To remove a computed expression, select it, and click **Remove** (GUI) or .

## Report Wizard: Sort By Panel

The **Sort By** panel specifies the field(s) you want to sort your report by.

In the GUI, from the **Available** list, select the field(s) to sort your report by and move them to the **Selected** list. You can change the sort order of a field by selecting it in the **Selected** list and clicking .

In the Web interface, click  to select the field(s) to sort your report by. You can change the sort order of a field by selecting it and clicking .

The sort order of each field is indicated by icons and tooltips.

## Report Wizard: Field Computations Panel

The **Field Computations** panel specifies aggregate computations for fields used as summary values in a report. *Aggregate computations* allow you to add numeric field values, calculate the average of multiple numeric field values, retrieve the smallest or largest numeric field value in a group of items, or count the number of items returned by an existing query. For example, if your report groups Project items by state and includes **Actual Budget** and **Estimated Budget** fields, you could display the totals for the **Actual Budget** and **Estimated Budget** fields in each state grouping.

---

**NOTE** The fields that display in the **Field Computations** panel must be selected in the **Item Fields** panel.

---

---

To create an aggregate expression, select a field, and click **Edit** in the GUI, or  in the Web interface. The **Edit Expression** dialog box displays. For more information, see “Add/Edit Expression Dialog Box” on page 126.

To remove an aggregate expression, select a field with an aggregate expression and click **Remove** (GUI) or  (Web).

To specify the format layout for an aggregate computation as it displays in the report, select the aggregate computation, and click **Format**. The **Format layout for field name** dialog box displays. For more information, see “Format Layout for field name Dialog Box” on page 126.

## Report Wizard: Change Package Type Panel

The **Change Package Type** panel specifies the change package type to display in a report.

From the **Change Package Type** list, choose a change package type to report on. By default, the available choices are: MKS Integrity and Implementer.

---

**NOTE** If your administrator has defined custom change package types, they display in the list. For more information, see your administrator.

---

## Report Wizard: Change Package Attributes Panel

The **Change Package Attributes** panel specifies change package fields to display in a report.

In the GUI, from the **Available** list, select the change package fields to display in the report and move them to the **Selected** list.

In the Web interface, click , select the change package fields to display in the report, and click **OK**.

The selected change package type determines the available fields.

## Report Wizard: Change Package Entry Attributes Panel

The **Change Package Entry Attributes** panel specifies change package entry attributes to display in a report.

Select the change package entry attributes to display in the report by doing the following:

- In the GUI, select the fields from the **Available** list and move them to the **Selected** list.
- In the Web interface, click  next to the **Change Package Attributes** field and use the data filter to select the fields.

---

**NOTE** The selected change package type determines the available entry attributes.

---

## Report Wizard: Attachment Fields Panel

The **Attachment Fields** panel specifies attachment fields to display in a report.

Select the attachment fields to display in the report by doing the following:

- In the GUI, select the fields from the **Available** list and move them to the **Selected** list.
- In the Web interface, click  next to the **Attachment Fields** field and use the data filter to select the fields.

---

To specify the format layout of an attachment field as it displays in the report, select an attachment field in the **Selected** column and click  A. The **Format layout for field name** dialog box displays. For more information, see “Format Layout for field name Dialog Box” on page 126. Formatted attachment fields are annotated with **(Formatted)** after the field name.

## Report Wizard: Attachment Attributes Panel

The **Attachment Fields** panel specifies attachment attributes to display in a report.

Select the attachment attributes to display in the report by doing the following:

- In the GUI, select the fields from the **Available** list and move them to the **Selected** list.
- In the Web interface, click  next to the **Attachment Attributes** field and use the data filter to select the fields.

Depending on the template you selected, a **Filter** field may display in this panel. Type the attachment types to include in the report, for example, \*.doc, report?\*.html, \*.txt, or \*.gif. Use commas to specify additional file types.

## Report Wizard: Test Result Fields Panel

The **Test Results Fields** panel specifies the test result fields to display in a report. A test result indicates the outcome of a test (pass, fail, other).

Select the test result fields to display in the report by doing the following:

- In the GUI, select the fields from the **Available** list and move them to the **Selected** list.
- In the Web interface, click  next to the **Test Result Fields** field and use the data filter to select the fields.

Filter test results by verdict or verdict type in the report by clicking in the **Filter By Verdict or Verdict Type** field (GUI) or the **Test Result Verdict Filter** field (Web) and using the data filter to select the verdicts and/or verdict types. To filter by verdict type, select All Pass, All Fail, or All Other. Verdicts are defined by your administrator, for example, Failed – Does Not Match Requirement. You can also filter for test results with no verdict by selecting Unspecified.

## Report Wizard: Test Step Results Panel

The **Test Step Results Fields** panel specifies the test step result fields to display in a report. Test steps are a special item type set up by your administrator that can be related to test case items.

Select the test step result fields to display in the report by doing the following:

- In the GUI, select the fields from the **Available** list and move them to the **Selected** list.
- In the Web interface, click  next to the **Test Step Results Fields** field and use the data filter to select the fields.

## Report Wizard: Test Result Attachment Attributes Panel

The **Test Result Attachment Attributes** panel specifies the attachment attributes to display in a report, for example, the attachment file name or the date it was added to the test result.

Select the test result attachment attributes to display in the report by doing the following:

- In the GUI, select the fields from the **Available** list and move them to the **Selected** list.

- 
- In the Web interface, click  next to the **Test Result Attachment Fields** field and use the data filter to select the fields.

Depending on the template you selected, a **Filter** field may display in this panel. Type the attachment types to include in the report, for example, \*.doc, report?\*.html, \*.txt, or \*.gif. Use commas to specify additional file types.

## Report Wizard: Test Result Related Items Panel

The **Test Result Related Items** panel specifies test result related item fields to display in a report. Related items display in sub-tables. Each sub-table contains its own set of table headers. The item fields for the root item do not display in the report.

Select which fields you want to include in your report:

-  specifies fields to add to the **Fields in Report** list using the data filter.
-  removes selected fields from the **Fields in Report** list.

To specify the format layout of a field as it displays in the report, select a field in the **Selected** column and click . The **Format layout for field name** dialog box displays. For more information, see “Format Layout for field name Dialog Box” on page 126. Formatted fields are annotated with (Formatted) after the field name.

## Report Wizard: Time Entry Fields Panel

The **Time Entry Fields** panel specifies time entry fields to display in a report.

Select the time entry fields to display in the report by doing the following:

- In the GUI, select the fields from the **Available** list and move them to the **Selected** list.
- In the Web interface, click  next to the **Time Entry Fields** field and use the data filter to select the fields.

## Report Wizard: Relationship Fields Panel

The **Relationship Fields** panel specifies relationship fields to display in a report.

From the **Available** list, select the relationship fields to display in the report and move them to the **Selected** list.

## Report Wizard: Relationship Filter Panel

The **Relationship Filter** panel specifies the criteria for an item to include in the related item set.

---

**NOTE** Field rules allow special characters, for example, (, ), [ , ], ", and '.

---

Creating a relationship filter is similar to creating an e-mail notification rule.

## Report Wizard: References Panel

After you create a report, the **References** panel displays all admin provided and user objects that reference the report. If you plan on making changes to the report or deleting it, this information lets

---

you know what objects are affected by the changes or deletion. For more information on admin provided objects, see the *MKS Integrity Server 2009 Administration Guide*.

---

**NOTE** This panel displays in the GUI only.

---

## Report Wizard: History Panel

After you create a report, the **History** panel displays the change history of the report, specifically outlining what report properties and values were added, deleted, or changed during each modification.

---

**NOTE** This panel displays in the GUI only.

---

## Add/Edit Expression Dialog Box

The **Add/Edit Expression** dialog box allows you to create or edit a computed expression used in a report. In a report, computed expressions can be used to do the following:

- If your report includes numeric fields, you can perform arithmetic calculations between one or more numeric fields and display the results in the report. For example, if your report currently displays a list of Project items, you could calculate the difference between the **Actual Budget** and **Estimated Budget** fields for each Project item and display the values in a new field called **Over/Under Budget**. The arithmetic calculation behind the **Over/Under Budget** is the *computed expression*.
  - If your report includes fields used as summary values, you can perform aggregate computations on these fields. *Aggregate computations* allow you to add numeric field values, calculate the average of multiple numeric field values, retrieve the smallest or largest numeric field value in a group of items, or count the number of items returned by an existing query. For example, if your report groups Project items by state and includes **Actual Budget** and **Estimated Budget** fields, you could display the totals for the **Actual Budget** and **Estimated Budget** fields in each state grouping.
- 1 In the **Expression Name** field, type a name for the computed expression as it will display in the report, for example, *Over/Under Budget*. If you do not define a name, the computed expression in the **Expression** field displays.
  - 2 In the **Expression** field, create a computed expression that incorporates numeric fields from the **Selected** list. For example, typing: ("Actual Budget" - "Estimated Budget") calculates the difference between the **Actual Budget** and **Estimated Budget** fields for each Project item and displays the values in the report.  
If you are creating an aggregate expression, typing `sum("Actual Budget")` adds the **Budget** fields in each state grouping of Project items to display a total actual budget by state.
  - 3 From the **Display Pattern** list, select or type a display pattern for the value of the computed expression as it will display in the report, for example, `$#,###.`
  - 4 To save the computed expression, click **OK**.

## Format Layout for field name Dialog Box

The **Format layout for field name** dialog box allows you to specify the format layout of a field as it displays in a report. The format layout you choose displays independent of the report styles selected in the **Style** panel.

---

You can specify alignment, font size, bold, italics, and colors.

## Report Wizard

Panel	Description
Type	<p>From the list, select a report type for your report. Each report type displays a description and preview.</p> <p><b>Note:</b> Depending on the report type you select, the following panels may or may not apply. In addition, the order of the panels may be slightly different.</p>
Attributes	<p>In the <b>Name</b> field, type a name for the report. This is the name of the report as it displays in the <b>Manage Reports</b> (GUI) or <b>Reports</b> (Web) view. A report name is mandatory and must be entered before you can move backward or forward in the wizard. You cannot have two reports with the same name.</p> <p>Next to the <b>Name</b> field,  displays, indicating that the report is a favorite. Favorites are MKS Integrity objects (queries, charts, reports, and dashboards) that you created and use.</p> <p>From the <b>Query</b> list, select a query to base the report on. The items returned by the query comprise the data in the report.</p> <p>In the <b>Description</b> field, describe the report.</p> <p>By default, report types and styles are loaded from the MKS Integrity Server when you create, edit, or run a report. When enabled, the <b>Use report type and style from the server</b> option ensures that any updates to the selected report type and styles are automatically reflected when you create, edit, or run the report. Clearing the <b>Use report type and style from the server</b> option stores the selected report type and styles with the report in the database. This ensures that the report maintains its current configuration if the selected report type and styles are updated on the MKS Integrity Server.</p> <p>To convert the report to an admin provided object, enable <b>Is Admin Provided</b>. Admin provided objects are objects that are used by your administrator to define the MKS Integrity solution for all users. For more information, see your administrator.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"><li>■ This option is available only if the <code>createSharedAdmin</code> or <code>Admin</code> permission is assigned to you.</li><li>■ Once a user object is converted to an admin provided object, you cannot revert it to a user object again.</li><li>■ Selecting <b>Is Admin Provided</b> automatically adds the currently logged in user to the Sharing tab with edit permissions.</li></ul> <p>If your report contains user defined date fields, you can change the default date format by selecting a format from the <b>Date Format</b> list or by typing your own.</p> <p>If your report contains user defined date fields that include the time, you can change the default date and time format by selecting a format from the <b>Date Time Format</b> list or by typing your own.</p>
Style	<p>In the GUI, do the following:</p> <ul style="list-style-type: none"><li>■ From the <b>Screen Style</b> list, choose a style that specifies how the report displays when viewed in a browser.</li><li>■ From the <b>Printer Style</b> list, choose a style that specifies how the report displays when printed.</li></ul> <p>In the Web interface, do the following:</p> <ul style="list-style-type: none"><li>■ To choose a style that specifies how the report displays when viewed in a browser, click <b>Screen Style</b>, then select a style from the <b>Style</b> list.</li><li>■ To choose a style that specifies how the report displays when printed, click <b>Printer Style</b>, then select a style from the <b>Style</b> list.</li></ul>

Panel	Description
<b>Logo</b>	<p>In the GUI, do one of the following:</p> <ul style="list-style-type: none"> <li>■ From the <b>Select Logo</b> list, choose an image file that resides on the MKS Integrity Server to display as a logo in the report.</li> <li>■ In the <b>Select Logo</b> field, type a URL to an image file. To clear the <b>Select Logo</b> field, delete the displayed logo or select the blank list item.</li> </ul> <p>In the Web interface, do one of the following:</p> <ul style="list-style-type: none"> <li>■ To choose a logo that resides on the MKS Integrity Server, click <b>Report Logo</b>, then select a logo from the list.</li> <li>■ To specify the URL of an image file, select <b>Specify the URL of the Logo</b> image, then type the URL.</li> <li>■ To display the image file after you type the URL, click <b>Show</b>. The image displays.</li> </ul> <p><b>Note:</b> Reports support GIF, JPEG, and PNG image types.</p>
<b>Parameters</b>	<p>The <b>Parameters</b> panel allows you to specify additional textual information in the report, for example, headers and footers.</p> <p>In the <b>Report Title</b> field, type the title of the report.</p> <p>In the <b>Report Header</b> field, type the header to display on each page of the report.</p> <p>In the <b>Report Footer</b> field, type the footer to display on each page of the report.</p> <p>Depending on the report type you selected, additional fields may display in the <b>Parameters</b> panel. These additional fields represent report items, such as labels, headings, and script variables that are used to define report segments. Segments are sub-sections of field information within a report. For example, if you select the <b>Detail - HTML, Column, Time Entry</b> report type, the report includes a segment that displays time entry field information. In the <b>Parameters</b> panel, a <b>Time Entry Heading</b> field displays the default name of the segment as <b>Time Entries</b>, which you can optionally change.</p>
<b>Sort By</b>	See "Report Wizard: Sort By Panel" on page 122.
<b>Item Fields</b>	See "To create a report" on page 120.
<b>Sharing</b>	<p>To select the principals (users and groups) you want to share your report with, click <b>Choose Principals</b> (GUI) or  (Web).</p> <p>To allow assigned principals to edit the query, select the principal's check box under the <b>Modify</b> column. To deny assigned principals the ability to edit the query, clear the principal's check box under the <b>Modify</b> column. By default, all assigned principals are denied the ability to edit a query.</p> <p><b>Note:</b> You cannot share a report to the <code>everyone</code> group unless you are an administrator or you have the <code>ShareToEveryone</code> permission. Contact your administrator for more information.</p>

---

# Editing a Report

---

**CLI EQUIVALENT** im editreport

---

You can edit the details of an existing report; however, you cannot change the report type. Changes made to the report affect the display of results the next time you run the report.

## **Key Considerations**

- If the report you are editing is based on a query that is invisible to you, the query displays in the **Query** list. However, this query is not available when you are creating a report.
- You can edit the reports that you created. Principals (users and groups) that a report is shared with can edit it if they have edit permissions assigned to them by the report creator.
- In addition to the panels specific to the report type, the following panels display in the GUI **Report Wizard**:
  - **References** displays all admin provided and user objects that reference the report. If you plan on making changes to the report or deleting it, this information lets you know what objects are affected by the changes or deletion.  
For more information on admin provided objects, see the *MKS Integrity Server 2009 Administration Guide*.
  - **History** displays all changes made to the report.

The following table describes how to edit a report:

Interface	Procedure
GUI	From the <b>Reports</b> view, select a report, then select <b>Properties</b> .
Web	From the <b>Reports View</b> , select a report, and select <b>Report Properties</b> .

# Copying a Report

---

**CLI EQUIVALENT** im copyreport

---

You can copy the details of a report to a new report and save the new report under a different name.

## **Key Considerations**

- If the report you are copying is an admin provided object, the **Is Admin Provided** option is cleared in the copy.
- A copied favorite (★) or non-favorite (☆) report becomes a favorite when you save the new report.
- Sharing information is not copied. You can copy sharing information by clicking **Copy Principals** on the **Sharing** tab.

---

**NOTE** You cannot share a report to the everyone group unless you are an administrator or you have the ShareToEveryone permission. Contact your administrator for more information.

---

The following table describes how to copy a report:

Interface	Procedure
GUI	From the <b>Manage Reports</b> view, select a report, and select <b>Report &gt; Copy</b> .
Web	From the <b>Reports View</b> , select a report, and select <b>Copy Report</b> .

# Running a Report

---

**CLI EQUIVALENT** im runreport

---

You can run a selected report or you can run a report for a selected item or set of items. For information on how to run an item report, see “Reporting on Items Using Item Reports” on page 134.

---

**NOTE** Displayed date fields do not change based on the time zone a user operates in; however, displayed date/time fields and time entries vary based on the time zone a user operates in.

---

The following table describes how to run a report:

Interface	Procedure
GUI	From the <b>Manage Reports</b> view, select a report, then select <b>Report &gt; Run</b> .
Web	From the <b>Reports View</b> , select a report, and select <b>Run Report</b> .
Hyperlink	Using a hyperlink, specify the following: <code>http://hostname:port/im/runreport?selection=reportname</code> where: <ul style="list-style-type: none"><li>■ <i>hostname</i> is the hostname of the MKS Integrity Server hosting the report</li><li>■ <i>port</i> is the port number of the of the MKS Integrity Server hosting the report</li><li>■ <i>reportname</i> is the name of the report</li></ul>

The report displays in a browser window.

Project details grouped by Product, ordered by State.

Mar 9, 2010

Item	Summary	State	Days in Current State	Assigned User	Risk	Effort Variance	Budget Variance
Product: Tempus Classic - (98)							
<a href="#">107</a>	Project for Release 1	Defined	161	jriley	High	-100%	
<a href="#">217</a>	Project for Release 2	Defined	36	jriley	Medium	-100%	
<a href="#">261</a>	Project for Hardware	Defined	129	jriley	Low	-100%	

Product Tempus Classic - (98) Item Total: 3

A report displaying all active projects grouped by Product (ascending order) and sorted by Health (descending order).

---

# Saving and Printing a Report

When you select the type of report you want to generate, the report displays in your default browser allowing you to save or print the report. You can also manipulate report data in Microsoft Excel.

The following tables describes how to save and print a report:

To...	Do this...
Save a report	With a report open in a browser window, select <b>File &gt; Save As</b> (Internet Explorer) or <b>File &gt; Save Page As</b> (Mozilla and Firefox).
Manipulate report data in Microsoft Excel	<p>Open the saved report in Microsoft Excel.</p> <p><b>Note:</b> If you are importing a CSV report, the context of fields in the report may require some manual configuration in Microsoft Excel.</p> <p>You can now use Microsoft Excel's report features, such as sorting and PivotTables, to manipulate the report data. You may save, export, or print the report data as with any other Excel table.</p>
Print a report	With a report open in a browser window, select <b>File &gt; Print</b> .

---

# Deleting a Report

---

**CLI EQUIVALENT** `im deletereport`

---

You can delete reports that are no longer needed from the database. Shared reports can be deleted only by the owner or an administrator.

The following table describes how to delete a report:

Interface	Procedure
GUI	From the <b>Manage Reports</b> view, select a report, then select <b>Report &gt; Delete</b> .
Web	From the <b>Reports View</b> , select a report, and select <b>Delete Report</b> .

# Reporting on Items Using Item Reports

You can run the following item reports:

- Report on a specified item or set of items, thus overriding the query specified in the report definition.
- Report on a specified item or set of items as of a date in the past (historical report).
- Report that compares the results of two reports from different time periods using a specified item or set of items (compare to base time report). See your administrator for more information.

## Running a Report on Selected Items

**NOTE** Running a report on a historical item runs the report as of the item's historical date. A dialog displays allowing you to choose a date. The item's historical date displays by default.

You can run reports on selected items in the GUI or in the Web interface.

Interface	To run a report on selected items...
GUI	Select one or more items, and select <b>Item &gt; Run Report</b> . The <b>Report Selection</b> dialog box displays. You can also run an item report through the <b>Relationships</b> view. If you are running a report on a historical item, select the report you want to run, and click <b>OK</b> . The report displays in your default browser. If the report contains item links and you edit the items listed in the report, you can update the open report by scrolling to the bottom and clicking <b>Re-run report</b> .
Web	In the Items view, select one or more items, click  and select <b>Run Item Report</b> . The <b>Run Report Against Specified Item(s)</b> page displays. Select the report you want to run, and click <b>OK</b> . The report displays in a new browser window.

## Running a Historical or Differences Report

A historical report is a report on a specified item or set of items as of a date in the past. An item differences report is a report that compares the results of two reports from different time periods using a specified item or set of items. Both reports are run from the **Run Report As Of** dialog.

### To run a historical report

1 Do one of the following:

- In the GUI, select one or more items, and select **Item > Historical > Run Report As Of**.
- In the Web interface, select one or more items, click  and select **Historical > Run Report As Of**.

The **Run Report As Of** dialog box displays.

**TIP** You can also run a historical report through the **Relationships View**.

2 From the list of reports, select the report you want to run.

- 
- 3 In the **Run Report As Of** box, select the applicable option from the list. You can run an item report based on date or edit. Depending on the selection, the box is populated with the corresponding date or edit selection options.

---

**IMPORTANT** If any item in the item selection was created after the selected start date, an error message displays when you run the report.

- 4 Click **OK**. If you selected to view a historical report, the HTML browser displays the report based on the option you selected.

### To run an item differences report

- 1 Do one of the following:
- In the GUI, select one or more items, and select **Item > Historical > Run Report As Of**.
  - In the Web interface, select one or more items, click  and select **Run item differences report**.
- The **Run Report As Of** dialog box displays.

---

**TIP** You can also run a compare to base time report through the **Relationships View**.

- 2 From the list of reports, select the report you want to run.
- 3 In the **Run Report As Of** box, select the applicable option from the list. You can run an item report based on date or edit. Your selection will be the base point in time that you want to compare the other report (and its base time) against. Depending on the selection, the box is populated with the corresponding date or edit selection options.
- 4 Select **Compare To Base Time**. Date is the default selection in the As Of drop list so a calendar displays allowing you to select the date of the report you want to compare. If you select another option, the corresponding date or edit selection options display.
- 5 Click **OK**. If you are comparing HTML reports, a report showing the results of the comparison displays in your default browser. Differences between the reports are highlighted.

If you are comparing XML reports, the **MKS Visual Difference** window displays the results of the comparison between the two reports. Differences between the reports are highlighted.

For more information on the MKS Visual Difference tool, see “MKS Visual Difference/Visual Merge Interfaces” on page 41.

## PART 5

# Viewing Project Data in Dashboards

---

# What is a Dashboard?

A *dashboard* is a static, user-definable view comprised of any combination of the following components: embedded charts, embedded reports, images, labels, and links to queries, reports, and Web sites.

Dashboards can be used to view any collection of components as a single unit, but are especially useful to provide a high-level overview of the status of a project or set of projects. You can design one dashboard for all projects and select the appropriate project(s) from the dashboard project filter at run time.

---

# Designing a Dashboard

You design dashboards by selecting the components you want and deciding where they will appear within the dashboard. When you run the dashboard, it displays the specified components according to the defined layout.

Using the dashboard layout designer, you can:

- create a set of grids and cells to represent the structure of the dashboard
- assign default properties to each grid and cell and override these properties for individual grids and cells
- add chart, report, link, label and image components into cells to build up the content that the user sees and interacts with
- define styles to be applied to text-based layout components
- easily highlight and change properties of grids, cells and cell elements
- preview a graphical representation of the dashboard's layout as it would appear in the GUI or Web interface

The types of components you can add to a dashboard are:

- **Charts**

You can embed any type of chart in a dashboard, and you can control the graph style of the chart as it appears in the dashboard. You can also specify the optimum width and height of the chart image as a means to manage the size of the dashboard. You can also specify whether chart results are filtered by the dashboard project filter.

- **Reports**

You can embed any type of report in a dashboard, and you can specify the optimum width and height of the report component as a means to manage the size of the dashboard. You can also specify whether report results are filtered by the dashboard project filter.

You can also link to a report using a link component.

- **Labels**

You can include text labels within the dashboard for elements such as titles, section headings, or descriptive text. You control the font and style of the label. You can also use empty labels to add space between other dashboard components.

- **Images**

You can specify an image file to display in a cell of the dashboard. This allows you to customize the dashboard with your company logo or some meaningful image representing the data presented on the dashboard. Only JPEG, PNG and GIF files can be used as images. To specify an image, you identify a local image file that is embedded directly in the dashboard.

Your administrator sets the maximum file size for dashboard images.

## ■ Links

Links allow you to access applications or information outside the dashboard. Links appear in the dashboard as named hyperlinks. You control the font and style of the link. Three types of links are supported:

- links to MKS Integrity reports, which open in separate browser windows
- links to MKS Integrity **Items** views using specific queries
- links to Web sites

For report and query links, you can specify whether the results are filtered by the dashboard project filter.

## **Example**

A project manager creates a dashboard to gain insight into a specific project using reports, queries, and charts.

The screenshot shows the TEMPUS Project Manager Dashboard. At the top, there's a header with the TEMPUS logo and the text "Project Manager". Below the header, a section titled "Project Manager Dashboard" is described as a template for project managers. It includes a "Reports and Queries" section with three categories: "Reports", "Queries", and "Links". The "Reports" category lists "All Projects Overview Report", "Active Projects by Product", and "Project Timeline Summary". The "Queries" category lists "Projects at Risk", "Active Projects", and "All Projects". The "Links" category lists "www.mks.com", "Project Management Institute (PMI)", and "Google". The main content area is titled "Project Summary" and features a "Project 107: Project for Release 1" card. This card displays project details like start date, acceptance date, plan date, specification freeze date, code freeze date, development completion, and project health. It also shows the state, classification, product, project manager, and total effort. Below this card is a "Process Metrics" section with a table showing pending, open, closed, and total change orders and defects. The dashboard has a dark theme with light-colored cards and tables.

---

# Dashboard Definition View

---

**CLI EQUIVALENT** `im viewdashboard`

---

The **Dashboard Definition** view displays the details of a dashboard definition.

---

**NOTE** You cannot display a dashboard definition in the Web interface.

---

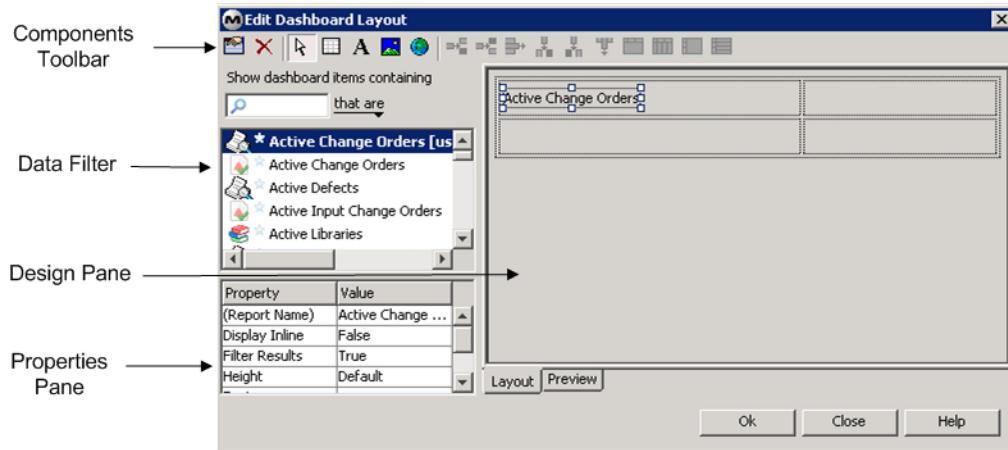
To display this view, select a dashboard in the **Manage Dashboards** view, and select **Dashboard > View Definition**.

To view general information for the dashboard, click a tab. To view layout details for the dashboard, click **View Layout**.

You cannot edit the dashboard definition while viewing it.

# Dashboard Layout Designer: User Interface Components

The following illustration shows the main features of the dashboard layout designer interface.



The dashboard layout designer incorporates drag-and-drop functionality to help you add components to a dashboard. You can drag a component from the data filter or add a component using one of the toolbar buttons in the components toolbar. You can modify the properties of specific components in the properties pane.

The Dashboard Layout Designer includes the following user interface components:

Component	Description
Components Toolbar	Includes tools to design a dashboard. For example: <ul style="list-style-type: none"><li>■ modify general properties and default styles for text, cells, and grids</li><li>■ select components of template, including grids, cells, and labels.</li><li>■ add new grid to dashboard</li><li>■ insert image into cell</li><li>■ inserts URL link into cell</li></ul> Certain tools can work in two modes: <ul style="list-style-type: none"><li>■ clicking the tool to operate in the selected mode, for example, clicking the text tool to insert a text label into a cell</li><li>■ dragging the tool onto the design pane to add the selected component, for example, dragging the image tool to insert an image into a cell</li></ul>
Data Filter	Allows you to locate and select objects to insert into the dashboard, such as charts, queries, and reports.
Design Pane	Area for designing your dashboard (using images, links, and tables).
Properties Pane	Displays properties for the selected object.

---

# Dashboards View: Managing Dashboards

---

**CLI EQUIVALENT** `im dashboards`

---

You can view and manage all queries created by you or shared to you from the **Manage Dashboards** view (GUI) or **Dashboards View** (Web interface).

To display the **Manage Dashboards** view in the GUI, select **Dashboard > Manage Dashboards**.

To display the **Dashboards View** in the Web interface, under **Quick Views**, click **Dashboards**.

In the GUI, the data filter in the **Manage Dashboards** view displays your favorite dashboards (dashboards that you created and use) by default. You can search for a specific dashboard by typing in the text filter and/or specifying a filter.

In this view, you can:

- create a dashboard (GUI only) (see “Creating a Dashboard” on page 143)
- edit a dashboard (GUI only) (see “Editing a Dashboard” on page 150)
- run a dashboard (see “Running a Dashboard” on page 151)
- delete a dashboard (GUI only) (see “Deleting a Dashboard” on page 154)
- copy a dashboard (GUI only) (see “Copying a Dashboard” on page 153)
- view a dashboard definition (see “Dashboard Definition View” on page 140)

# Creating a Dashboard

---

**CLI EQUIVALENT** `im createdashboard`

---

You create a dashboard through the GUI.

You can also create a dashboard directly in XML. The XML file must conform to a specific document type definition (DTD). For more information, see “Creating a Dashboard in XML” on page 155.

## Key Considerations

- You can define a project filter to apply to a dashboard at run time. Depending on how you design your dashboard, the dashboard project filter may or may not be applied to a dashboard component. It is important to understand how dashboard and project filters work together.
- Embedded reports require a lot of memory when the dashboard is run. Using multiple embedded reports in a dashboard can affect performance.
- To select charts, queries, reports, principals, projects, fields, and field values in the **Create Dashboard** and **Edit Dashboard Layout** dialog boxes, you must be familiar with using the data filter.

## To create a dashboard

- 1 Select **Dashboard > Create**. The **Create Dashboard** dialog box displays.
- 2 Provide information for the **Name** field, and the **Description**, **Sharing**, and **Project Filter** tabs. For more information, see “Dashboard Properties” on page 145.
- 3 To begin designing your dashboard, click **Edit Layout**. The **Edit Dashboard Layout** dialog box displays.
- 4 To set global layout properties, click . The **Layout Properties** dialog box displays.

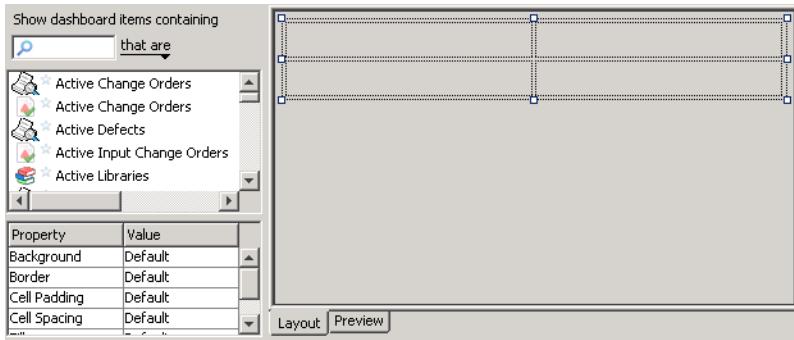
---

**TIP** By setting global layout properties, you can avoid having to configure detailed settings for each individual element of your dashboard and instead have a consistent style that is applied to all instances of a text, cell, or grid element. All global dashboard properties can be overridden for individual components.

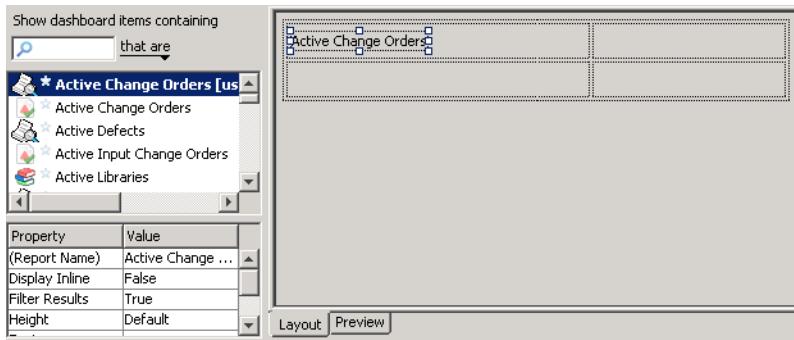
---

For detailed information on the fields in the **Layout Properties** dialog box, see the following:

- “Editing a Dashboard” on page 150
  - “Dashboard Layout Properties: General Tab Fields” on page 147
  - “Dashboard Layout Properties: Text Styles Tab Fields” on page 149
  - “Dashboard Layout Properties: Default Text Styles Tab Fields” on page 147
  - “Dashboard Layout Properties: Default Grid Table Fields” on page 148
  - “Dashboard Layout Properties: Default Cell Tab Fields” on page 148
- 5 To add a new grid to your dashboard, drag the grid tool onto the design pane. The **New Grid** dialog box displays.
  - 6 Set the size of the grid by entering the number of rows and columns, and click **OK**. The new grid is inserted on the design pane. The default properties of the grid display in the properties pane.



- 7 In the properties pane, modify the properties of the grid as required by clicking the **Value** column beside the property you want to change. For detailed information on grid property options, see “Dashboard Layout Properties: Default Grid Table Fields” on page 148.
- 8 To add an embedded chart, an embedded report or report link, or a query link to the dashboard, select the appropriate type from the components filter list. The components view pane displays the components associated with the selection that are available to you. For example, if you specified a chart component type, all charts created by you or shared with you display in the components view pane.
- 9 Select a component, and drag it into the target cell of the grid. The default properties for the component display in the properties pane.



#### **NOTE**

- If you decide to leave empty cells in the grid, empty space displays in the dashboard for both the GUI and Web interface.
- When you insert a chart in the grid, it shows at its requested size with a placeholder image and border. Actual chart data can be viewed by previewing the chart.

- 10 In the properties pane, modify the properties of the component as required. For detailed information on the properties for charts, reports, and queries, see the following:
  - “Dashboard Report Properties” on page 149
  - “Dashboard Chart Properties” on page 147
  - “Dashboard Query Link Properties” on page 148
- 11 Repeat the previous steps as required to insert the desired charts, reports, and queries into the grid.
- 12 Add labels, images, or URLs to the dashboard using the toolbar buttons. The default properties of the label, image, or link display in the properties pane.

- 
- 13** Edit the properties of the labels, images, or URLs as required. For detailed information on the properties for these components see the following:
- “Dashboard Label Properties” on page 147
  - “Dashboard Image Properties” on page 147
  - “Dashboard URL Properties” on page 149
- 14** Edit individual cells in the grid by selecting the cell and editing the properties as required. For detailed information on the properties for grid cells see “Dashboard Layout Properties: Default Cell Tab Fields” on page 148.
- 15** To preview your changes, click the **Preview** tab. If any charts or reports are embedded in the dashboard, they run for the preview. Depending on how you designed your dashboard, embedded charts and reports may or may not be filtered by any currently applied dashboard project filter.
- 16** To save the dashboard layout, click **OK** on the **Edit Dashboard Layout** dialog box. The **Create Dashboard** dialog box displays.
- 17** To save the dashboard, click **OK** on the **Create Dashboard** dialog box.

## Dashboard Properties

Field/Tab	Description
<b>Name</b>	Type a name for the new dashboard. The name you choose should help you associate the dashboard with the type of information it presents. Next to the <b>Name</b> field,  displays, indicating that the dashboard is a <i>favorite</i> . Favorites are MKS Integrity objects (queries, charts, reports, and dashboards) that you created and use. To convert the dashboard to an admin provided object, enable <b>Is Admin Provided</b> . Admin provided objects are objects that are used by your administrator to define the MKS Integrity solution for all users. For more information, see your administrator. <b>Note:</b> <ul style="list-style-type: none"> <li>■ This option is available only if the <code>createSharedAdmin</code> or <code>Admin</code> permission is assigned to you.</li> <li>■ Once you convert a user object to an admin object, you cannot revert it to a user object again.</li> <li>■ Selecting the <b>Is Admin Provided</b> option automatically adds the currently logged in user to the <b>Sharing</b> tab with edit permissions.</li> </ul>
<b>Edit Layout</b>	Defines the layout of the dashboard. For more information, see “To create a dashboard” on page 143.
<b>Description</b>	Type a description of the dashboard

Field/Tab	Description
<b>Sharing</b>	<p>Click <b>Choose Principals</b> and select the principals (users and groups) you want to share your dashboard with.</p> <p>To allow assigned principals to edit the dashboard, select the principal's check box under the <b>Modify</b> column. To deny assigned principals the ability to edit the dashboard, clear the principal's check box under the <b>Modify</b> column. By default, all assigned principals are denied the ability to edit a dashboard.</p> <p><b>Note:</b> You cannot share a dashboard to the <code>everyone</code> group unless you are an administrator or you have the <code>ShareToEveryone</code> permission. Contact your administrator for more information.</p>
<b>Project Filter</b>	<p>Select the <b>Filter Type</b> to specify how the dashboard can be filtered when it is run. Choose one of the following:</p> <ul style="list-style-type: none"> <li>■ <b>Open</b> allows all projects to be selected as filter values when the dashboard is run. You can also select default filter values to apply to the dashboard at run time.</li> <li>■ <b>Fixed</b> sets the project filter values that are always applied when the dashboard is run. If this option is selected, you cannot modify the project filter at run time.</li> <li>■ <b>Restricted</b> allows users to only select from a defined list of project filter values when the dashboard is run. You can also select default filter values from the restricted set to apply to the dashboard at run time.</li> <li>■ <b>None</b> prevents users from filtering the dashboard. The project filter does not display at run time.</li> </ul> <p><b>Note:</b> Depending on how you design your dashboard layout, the dashboard filter may not be applied to chart, report, or query dashboard components.</p> <p>If you selected the <b>Fixed</b> or <b>Restricted</b> filter type, click  in the <b>Available Projects</b> list and select the projects that are always used to filter the dashboard data (Fixed) or the projects that can be selected from when the dashboard is run (Restricted). To remove a project from the <b>Available Projects</b> list, select it and click .</p> <p>If you selected the <b>Open</b> or <b>Restricted</b> filter type, click  in the <b>Default Selected Projects</b> list and select the projects to use as the default project filter when the dashboard is run. You can override the default at run time. To remove a project from the <b>Default Selected Projects</b> list, select it and click .</p>

## Dashboard Chart Properties

Field	Description
<b>Chart Name</b>	Name of chart. Cannot edit value.
<b>Filter Results</b>	Whether chart data is filtered by dashboard project filter. If <code>false</code> , dashboard project filter is ignored.
<b>Graph Style</b>	Style of graph used for chart when run in dashboard. Can be different from graph style specified when chart was designed, but must be valid for chart type.
<b>Height</b>	Height to scale chart to. <b>Note:</b> Option ignored for charts using table graph style.
<b>Width</b>	Width to scale chart to. <b>Note:</b> Option ignored for charts using table graph style.

## Dashboard Label Properties

Field	Description
<b>Text</b>	Label text.
<b>Text Style</b>	Style applied to label. Select from Default, Heading, or any custom styles defined as global layout property.

## Dashboard Image Properties

Field	Description
<b>Height</b>	Height in pixels to scale image to.
<b>Image Data</b>	Click the browse button to choose the image file. JPEG, PNG, and GIF formats supported. Size of selected image file displays in column. Administrator sets maximum size for dashboard image. <b>Note:</b> Image file types are case-sensitive. MKS Integrity recognizes *.gif, *.jpeg, *.jpg, and *.png file types only.
<b>Width</b>	Width in pixels to scale image to.

## Dashboard Layout Properties: General Tab Fields

Field	Description
<b>Background Color</b>	Preferred background color of dashboard from: <ul style="list-style-type: none"><li>■ available named HTML color</li><li>■ Swatches, HSB (Hue, Saturation, Brightness), or RGB (Red, Green, Blue).</li></ul> By default, background color is <code>default</code> as determined by operating system.

## Dashboard Layout Properties: Default Text Styles Tab Fields

Field	Description
<b>Default Label</b>	By default, label text style is <code>system</code> .
<b>Default Report Link</b>	By default, report link text style is <code>system</code> .
<b>Default Query Link</b>	By default, query link text style is <code>system</code> .
<b>Default URL Link</b>	By default, URL link text style is <code>system</code> .

## Dashboard Layout Properties: Default Grid Table Fields

Default Grid Properties	Description
<b>Background Color</b>	Background color of grid selected from: <ul style="list-style-type: none"><li>■ available color</li><li>■ custom option for Swatches, HSB (Hue, Saturation, Brightness), or RGB (Red, Green, Blue).</li></ul> OS determines default background color.
<b>Border</b>	Thickness of border around grid from 0 to 99 pixels. By default, border is 0.
<b>Cell Spacing</b>	Distance between individual cells from 0 to 99 pixels. By default, cell spacing is 0.
<b>Cell Padding</b>	Distance between cell's contents and cell margin from 0 to 99 pixels. By default, cell padding is 0.
<b>Pack</b>	Whether any extra horizontal space in grid is filled by last column (at the far right). If pack is <code>False</code> then extra horizontal space is distributed across all columns. Extra horizontal space only occurs when <code>Fill</code> option is <code>True</code> . By default, pack is <code>True</code> . <b>Note:</b> Resizable components such as embedded reports, wrapping text labels, and charts using table graph style take on their minimum size when placed in packed cell. In general, these types of components should not be placed in packed cells.
<b>Fill</b>	Whether grid expands to fill available width in dashboard display. If fill is <code>False</code> grid uses only space required to display its contents. By default, fill is <code>True</code> .

## Dashboard Layout Properties: Default Cell Tab Fields

Fields	Description
<b>Background Color</b>	Preferred background color of cell selected from: <ul style="list-style-type: none"><li>■ available named HTML color</li><li>■ custom option for Swatches, HSB (Hue, Saturation, Brightness), or RGB (Red, Green, Blue)</li></ul> OS determines default background color.
<b>Horizontal Alignment</b>	Horizontal alignment of cell contents ( <code>Left</code> , <code>Center</code> , or <code>Right</code> ). By default, horizontal alignment is <code>Left</code> .
<b>Vertical Alignment</b>	Vertical alignment of cell contents ( <code>Top</code> , <code>Middle</code> , or <code>Bottom</code> ). By default, vertical alignment is <code>Middle</code> .
<b>Wrap</b>	Wrapping of text within cell. By default, <code>Wrap</code> is enabled.

## Dashboard Query Link Properties

Field	Description
<b>Query Name</b>	Name of query. Cannot edit value.
<b>Filter Results</b>	Whether query results are filtered by dashboard project filter. If <code>false</code> , dashboard project filter is ignored.
<b>Text</b>	Text used as link name in place of query name. Used to give more meaningful name to link if required.
<b>Text Style</b>	Style applied to query name or text. Select from <code>Default</code> , <code>Heading</code> , or any custom styles defined as global layout property.

## Dashboard Report Properties

Field	Description
<b>Report Name</b>	Name of report. Cannot edit value.
<b>Display Inline</b>	Whether report is embedded in dashboard or linked to from dashboard.
<b>Filter Results</b>	Whether report data is filtered by dashboard project filter. If <code>false</code> , dashboard project filter is ignored.
<b>Height</b>	Maximum height of report if displayed inline (see Display Inline option).
<b>Text</b>	Text used as link name in place of report name. (Used to give more meaningful name to link if required.)
<b>Text Style</b>	Style applied to report name or text. Select from Default, Heading, or any custom styles defined as global layout property.
<b>Width</b>	Maximum width of report if displayed inline (see Display Inline option).

## Dashboard URL Properties

Field	Description
<b>Text</b>	Text used as link name in place of URL. Used to give a more meaningful name to the link if required.
<b>Text Style</b>	Style applied to URL or text. Select from Default, Heading, or any custom styles defined as a global layout property.
<b>URL</b>	URL to launch in browser, for example, <a href="http://www.mks.com">http://www.mks.com</a> .

## Dashboard Layout Properties: Text Styles Tab Fields

Fields	Description
<b>Defined Text Styles</b>	By default, styles are defined for Heading. Heading style is available to apply to any text you define as heading. To modify defined text style, select it, and change its characteristics, and then click <b>Apply</b> . To create new text style, click <b>New</b> .
<b>Name</b>	Name of defined text style you select. By default, first one is selected.
<b>Font</b>	Choose font for selected text style. OS determines default font.
<b>Size</b>	Choose available font sizes from 8–72 points. Also choose font weight, such as Bold, or Italic, or both. OS determines default font size.
<b>Color</b>	Choose font color from available system font colors. OS determines default color.
<b>Sample</b>	Sample of text, including font size, weight, and color.

---

# Editing a Dashboard

---

**CLI EQUIVALENT** im editdashboard

---

In the GUI, you can edit an existing dashboard.

In addition to the tabs specific to the dashboard, the following tabs display in the **Edit Dashboard** dialog box:

- **References** displays all admin provided and user objects that reference the dashboard. If you plan on making changes to the dashboard or deleting it, this information lets you know what objects are affected by the changes or deletion.

For more information on admin provided objects, see the *MKS Integrity Server 2009 Administration Guide*.

- **History** displays all changes made to the dashboard.

To edit a dashboard, select a dashboard from the **Manage Dashboards** view, then select **Dashboard > Edit**.

# Running a Dashboard

**CLI EQUIVALENT** im rundashboard

Dashboards can be run in the GUI or Web interface.

## Key Considerations

- If the dashboard includes a component for an underlying object that is invisible to you, the component displays in the dashboard. If the underlying object for a dashboard component is removed from the system, the dashboard displays an error in place of the component.
- Charts using the table graph style are not sized according to the height and width properties specified in the dashboard definition.
- Displayed date fields do not change based on the time zone a user operates in; however, displayed date/time fields and time entries vary based on the time zone a user operates in.

The following table describes how to run a dashboard:

Interface	Procedure
GUI	From <b>Manage Dashboards</b> view, select a dashboard, and select <b>Dashboard &gt; Run</b> .
Web interface	From <b>Dashboards View</b> , select a dashboard, and select <b>Run Dashboard</b> .

While a dashboard is running, you can perform the following activities in the dashboard:

Dashboard Activity	Description
Apply a Project Filter to the Dashboard	<p>The projects that display when you run the dashboard are based on the project filters defined for the dashboard.</p> <p>Depending on how you design your dashboard, the dashboard project filter may or may not be applied to the dashboard components. If the dashboard project filter is not applied to the dashboard components, <b>Project Filter</b> is disabled in the <b>Dashboard</b> view.</p> <p>You can change the project selection (except when the dashboard is defined with a fixed project filter or no project filter). If you want to filter further within the projects in the <b>Project Filter</b>, select one or more of the displayed projects.</p> <p>If the dashboard project filter is applied to a report, report link, or query link component, the dashboard project filter is applied in addition to any filter defined for the component. If the dashboard project filter is applied to a chart component, the dashboard project filter overrides any project filter defined for the chart.</p> <p>To use the project filter, you must be familiar with using the data filter.</p> <p><b>Tip:</b> When running a dashboard in the GUI, you can toggle the display of the project filter panel by clicking the arrows beside <b>Project Filter</b>.</p>
Refresh the Dashboard	When you refresh a dashboard, the contents of any embedded charts or reports are updated. To refresh the contents of a dashboard in the GUI, click <b>View &gt; Refresh</b> . To refresh the contents of the chart in the Web interface, click <b>Refresh View</b> .

Dashboard Activity	Description
Click a Query Link	<p>Clicking a query link in a dashboard opens the <b>Items</b> view and displays items based on the query filter. Depending on how you designed your dashboard, any current dashboard project filter may also be applied to the query results. In the GUI, the view displays in a separate window; in the Web, the view replaces the dashboard.</p> <p>You cannot change the project filter while viewing items for a query link. If you have set up a project filter preference for the <b>Items</b> view, it is disregarded until you run another query.</p>
Click Chart Values	<p>Clicking a value in a distribution chart in a dashboard opens the <b>Items</b> view and displays items that make up the chart value. In the GUI, the view displays in a separate window; in the Web, the view replaces the dashboard.</p> <p>For item fields charts and item fields trend charts, if you want to view a single item that makes up a particular chart value, click the chart value (either an area of a bar, pie, or trend style chart, or a cell in a table style chart). In the GUI, the <b>Item Detail</b> view displays detailed information about the item that makes up that portion of the chart's data set. In the Web, the view replaces the dashboard.</p> <p>You cannot change the project filter while viewing items for a chart segment. If you have set up a project filter preference for the <b>Items</b> view, it is disregarded until you run another query.</p>
Click Report Link	<p>Clicking a report link in a dashboard opens a separate browser window containing the results of the report. Depending on how you designed your dashboard layout, any current dashboard project filter may also be applied to the report results.</p>
Click URL Link	<p>Clicking a URL link in a dashboard opens a separate browser window targeted at the URL as specified in the dashboard definition.</p>
Resize Charts	<p>In the GUI, you can resize a bar or pie chart in the dashboard by right-clicking and selecting <b>Zoom In</b> or <b>Zoom Out</b>. For bar charts, you also need to select the axis to resize. The axes you can select depend on the graph style of the chart. Select <b>Auto Range</b> to restore the chart to its original size. In the Web interface, you can zoom out on a bar chart's axes by clicking <b>Zoom Out</b> and zoom in by clicking <b>Zoom In</b>. For pie charts, these options are disabled.</p> <p>To increase the size of the chart in the Web interface, click <b>Increase Size</b>, and to decrease the size of the chart, click <b>Decrease Size</b>.</p>
Sort Tables	<p>You can sort a table chart in the dashboard by column or row. Click a column to sort the rows in ascending or descending order for the values in that column; click a row to sort the columns in ascending or descending order for the values in that row.</p>

---

# Copying a Dashboard

---

**CLI EQUIVALENT** `im copydashboard`

---

In the GUI, you can copy a dashboard to use it as the basis for a new dashboard. For example, you can create a generic dashboard to reflect your corporate standards, and use that dashboard as the basis for all other dashboards.

To copy a dashboard in the GUI, select the dashboard from the **Manage Dashboards** view, then select **Dashboard > Copy**.

## **Key Considerations**

- A copied favorite (★) or non-favorite (☆) dashboard becomes a favorite when you save the new dashboard.
- If the dashboard you are copying is an admin provided object, the **Is Admin Provided** option is cleared in the copy.
- Sharing information is not copied. You can copy sharing information by clicking **Copy Principals** on the **Sharing** tab.

---

**NOTE** You cannot share a dashboard to the `everyone` group unless you are an administrator or you have the `ShareToEveryone` permission. Contact your administrator for more information.

---

---

# Deleting a Dashboard

---

**CLI EQUIVALENT** `im deletedashboard`

---

If you decide you no longer want to use a dashboard, you can delete it in the GUI. You can only delete dashboards that you created, unless you have administrator permissions.

The following table describes how to delete a dashboard:

Interface	Procedure
GUI	From the <b>Manage Dashboards</b> view, select a dashboard, and select <b>Dashboard &gt; Delete</b> .
Web	From the <b>Dashboards View</b> , select a dashboard, and select <b>Delete Dashboard</b> .

---

# Creating a Dashboard in XML

When you create a dashboard using the MKS Integrity dashboard designer (see “Creating a Dashboard” on page 143), MKS Integrity saves your dashboard layout files in XML format in the database. If you prefer to create a dashboard directly in XML without using the dashboard designer, it must conform to a specific document type definition (DTD).

If you create a dashboard directly in XML, you may not be able to edit it using the MKS Integrity dashboard designer.

The content for the Dashboard DTD is as follows:

```
<!-- DashboardLayout is the root element for the layout definition -->
<!ELEMENT DashboardLayout (Styles?,Grids?)>
<!ATTLIST DashboardLayout backgroundColor CDATA #IMPLIED>
<!ATTLIST DashboardLayout version CDATA #IMPLIED>

<!--Styles contain the user defined text styles and the default style assignments
for the layout-->
<!ELEMENT Styles (TextStyle*,DefaultStyles?)>

<!--TextStyle defines a user defined style-->
<!ELEMENT TextStyle EMPTY>
<!ATTLIST TextStyle fontColor CDATA #IMPLIED>
<!ATTLIST TextStyle fontName CDATA #IMPLIED>
<!ATTLIST TextStyle fontSize CDATA #IMPLIED>
<!ATTLIST TextStyle fontStyle ( bold | italic | bolditalic | plain ) "plain">
<!-- The id of the text style is how other elements reference this text style,
currently should be the same as the name-->
<!ATTLIST TextStyle id id #IMPLIED>
<!ATTLIST TextStyle name CDATA #REQUIRED>

<!--DefaultStyles defines what styles (by id) are to be assigned to what
elements-->
<!ELEMENT DefaultStyles EMPTY>
<!ATTLIST DefaultStyles label idref #IMPLIED>
<!ATTLIST DefaultStyles querylink idref #IMPLIED>
<!ATTLIST DefaultStyles reportlink idref #IMPLIED>
<!ATTLIST DefaultStyles urllink idref #IMPLIED>

<!--Grids is the container for all Grid elements-->
<!ELEMENT Grids (DefaultGridProperties?,DefaultCellProperties?,Grid*)>

<!--DefaultGridProperties define the default attributes of all grids-->
<!ELEMENT DefaultGridProperties EMPTY>
<!ATTLIST DefaultGridProperties backgroundColor CDATA #IMPLIED>
<!ATTLIST DefaultGridProperties border CDATA #IMPLIED>
<!ATTLIST DefaultGridProperties cellpadding CDATA #IMPLIED>
<!ATTLIST DefaultGridProperties cellspacing CDATA #IMPLIED>
<!--fill set to true means that the grid should occupy all horizontal space in
the layout-->
<!ATTLIST DefaultGridProperties fill ( True | False ) "True">
<!--pack set to true means that all but the last grid cell in a row should take
on the preferred size of their contents CellElements-->
<!ATTLIST DefaultGridProperties pack ( True | False ) "False">

<!--DefaultCellProperties define the default attributes of all cells-->
<!ELEMENT DefaultCellProperties EMPTY>
<!ATTLIST DefaultCellProperties align ( Left | Center | Right ) "Left">
<!ATTLIST DefaultCellProperties backgroundColor CDATA #IMPLIED>
<!ATTLIST DefaultCellProperties valign ( Top | Middle | Bottom ) "Middle">
<!ATTLIST DefaultCellProperties wrap ( True | False ) "True" >
```

---

```

<!--Grid is a container for cells. A layout can have multiple grid instances.-->
<!ELEMENT Grid ((Cell+)*)>
<!ATTLIST Grid backgroundColor CDATA #IMPLIED>
<!ATTLIST Grid border CDATA #IMPLIED>
<!ATTLIST Grid cellpadding CDATA #IMPLIED>
<!ATTLIST Grid cellspacing CDATA #IMPLIED>
<!ATTLIST Grid fill ( True | False )>
<!ATTLIST Grid pack ( True | False )>

<!--
Cell contains the actual dashboard layout components that the user sees and
interacts with.
Cells are organized into a grid with the top left cell being at x=0, y=0. A cell
may span more than one row or column by specifying a width or height > 1
-->
<!ELEMENT Cell ((Label | URLLink | ReportLink | QueryLink | Chart | Image )*)>
<!ATTLIST Cell align ( Left | Center | Right )>
<!ATTLIST Cell backgroundColor CDATA #IMPLIED>
<!ATTLIST Cell height CDATA #IMPLIED>
<!ATTLIST Cell valign ( Top | Middle | Bottom )>
<!ATTLIST Cell width CDATA #IMPLIED>
<!ATTLIST Cell wrap ( True | False )>
<!ATTLIST Cell x CDATA #REQUIRED>
<!ATTLIST Cell y CDATA #REQUIRED>

<!--Label allows the designer to include descriptive text in the dashboard
layout. This text will wrap if the cell attributes allow wrapping.-->
<!ELEMENT Label EMPTY>
<!ATTLIST Label text CDATA #IMPLIED>
<!ATTLIST Label textStyle idref #IMPLIED>

<!--URLLink allows the designer to include a link that will launch a browser to
the given hyperlink reference-->
<!ELEMENT URLLink EMPTY>
<!--href is the URL definition of the link (e.g. http://www.mks.com)-->
<!ATTLIST URLLink href CDATA #IMPLIED>
<!ATTLIST URLLink text CDATA #IMPLIED>
<!ATTLIST URLLink textStyle idref #IMPLIED>

<!--ReportLink allows the designer to include a link that will launch an
Integrity Manager report.-->
<!ELEMENT ReportLink EMPTY>
<!--name is the fully qualified name of the report as defined in the Integrity
Manager application-->
<!ATTLIST ReportLink name CDATA #IMPLIED>
<!ATTLIST ReportLink text CDATA #IMPLIED>
<!ATTLIST ReportLink textStyle idref #IMPLIED>
<!ATTLIST ReportLink filtered ( True | False )>

<!--QueryLink allows the designer to include a link that will launch an Integrity
Manager items view against the given query.-->
<!ELEMENT QueryLink EMPTY>
<!--name is the fully qualified name of the query as defined in the Integrity
Manager application-->
<!ATTLIST QueryLink name CDATA #IMPLIED>
<!ATTLIST QueryLink text CDATA #IMPLIED>
<!ATTLIST QueryLink textStyle idref #IMPLIED>
<!ATTLIST QueryLink filtered ( True | False )>

<!--Chart allows the designer to include an Integrity Manager chart in the
dashboard layout.-->
<!ELEMENT Chart EMPTY>
<!--name is the fully qualified name of the chart as defined in the Integrity
Manager application-->
<!ATTLIST Chart name CDATA #IMPLIED>

```

---

```
<!--graphStyle allows the dashboard designer to override the style of chart at
run time. The default will be whatever the chart designer requested.-->
<!ATTLIST Chart graphStyle ( VerticalBar | VerticalStackedBar | HorizontalBar |
HorizontalStackedBar | Pie | Line | Table )>
<!ATTLIST Chart height CDATA #IMPLIED>
<!ATTLIST Chart width CDATA #IMPLIED>
<!ATTLIST Chart filtered ( True | False )>

<!--Image allows the designer to include graphics in the dashboard layout.-->
<!ELEMENT Image (ImageData)>
<!ATTLIST Image height CDATA #IMPLIED>
<!ATTLIST Image width CDATA #IMPLIED>

<!--ImageData is the element that actually contains the image source. The source
must be a base 64 (see http://www.ietf.org/rfc/rfc2045.txt) encoded gif, jpg or
png image.-->
<!ELEMENT ImageData (#CDATA)>
```

## PART 6

# Performing Calculations Between Fields

# What are Computed Expressions?

MKS Integrity can perform simple or complex calculations of item data in fields, charts, and reports using computed expressions. A *computed expression* is any combination of mathematical or logical operators, constants, functions, and MKS Integrity fields that evaluates to a single value. This functionality offers more power and flexibility to your MKS Integrity database by dynamically producing specific information about your item data.

You can perform calculations in fields, charts, and reports by creating two types of computed expressions: intra-item or inter-item.

## Intra-item Computed Expressions

*Intra-item computed expressions* perform calculations between MKS Integrity fields in a single item, storing the result in a read-only *computed field*. The computed field also stores the computed expression. For example, you could create an intra-item expression that adds the values of the **QA Estimated Time** and **Development Estimated Time** fields in a Feature item to produce a value in the **Total Estimated Time** field (the computed field).

Computed fields are created by administrators in the MKS Integrity Administration Client. To users, computed fields display as read-only fields in items; however, administrators can configure the field visibility. Administrators can also configure how and when computed fields calculate.

Intra-item expressions can also retrieve information from an item using *external information functions*. For example, the `CPECount()` external information function counts the number of change package entries in an item. External information functions are useful for retrieving workflow data, for example, how long a Customer Request item has been in a state of `Submit`.

The screenshot shows two windows side-by-side. On the left is the 'Computation Values' dialog, which contains a 'Computation Definition' section with a code editor containing the expression: `IsEmpty("Pending Change Order Count",0) + IsEmpty("Pending Defect Count",0) + IsEmpty("Pending Test Session Count",0)`. Below this are settings for 'Store to History Frequency' (daily), 'How to Run Computations' (static), and a 'Compute Now' button. A note at the bottom says 'Last Evaluation At: Wed Sep 15 02:01:05 EDT 2010'. On the right is an 'Item Details' view for a 'Product' item (ID: 98). The 'Tasks' section shows a table with columns: Pending, Open, Closed, Total. The 'Tasks' row has a value of 13. The 'Work Items' row has a value of 0. Other rows include Change Orders (0), Defects (13), Test Objectives (0), Test Sessions (11), and Change Packages (0). Below this is a 'Projects' section with a table for 'At Risk' projects, showing two entries: ID 104 (Project, State: Defined, Assigned User: project\_ma..., Created By: administrator 1) and ID 477 (Project, State: Defined, Assigned User: project\_ma..., Created By: administrator 2).

The computed expression for a computed field in the MKS Integrity Administration Client (left) and the computed field as it displays in the Item Details view of an item (right).

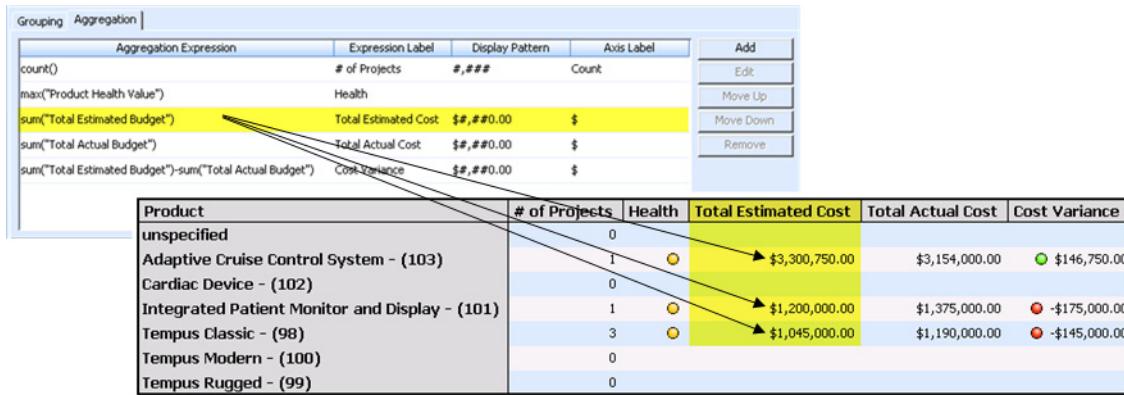
Once you define an intra-item expression using external information functions, you can use queries, charts, reports, and dashboards to collect the metric data and report on it.

## Inter-item Computed Expressions

*Inter-item computed expressions* (or *aggregate computed expressions*) perform calculations against MKS Integrity fields in a list of MKS Integrity items. To do this, you create a computed expression using an aggregate function, such as `sum` or `average`. For example, you could create an inter-item

expression in a report that uses the `sum(field)` aggregate function to add the **Estimated Cost** field in a list of Project items and produce a value in a new **Total Estimated Cost** field in the report.

Aggregate expressions can be created in a chart or report by any user in the MKS Integrity Client or MKS Integrity Administration Client. Inter-item computed expressions are useful for generating aggregate data from MKS Integrity fields in a report or chart.

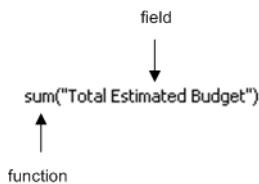


*The aggregate expression in a chart's properties (left) and the results of the computation in the generated chart (right).*

# Computed Expression Rules

The rules for writing computed expressions are similar to the rules for writing expressions in most programming languages. Computed expressions use a specific syntax and operators, constants, functions, and MKS Integrity fields to perform calculations.

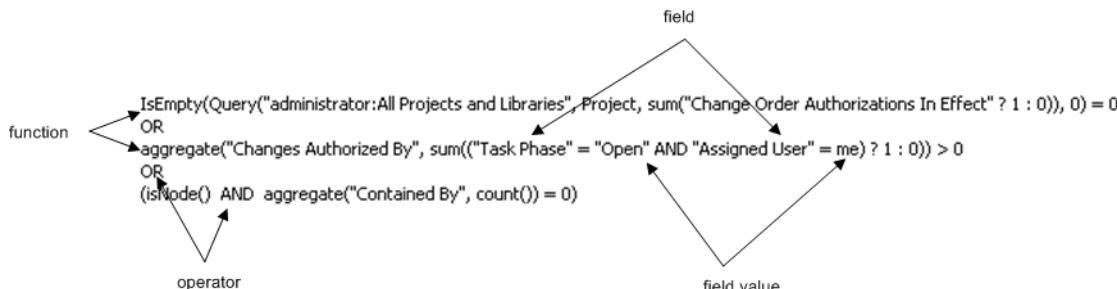
For example, the following diagram displays a simple computed expression involving one function and one field to perform an aggregate calculation:



The computed expression is used in a chart to calculate the sum total of a field that appears in several items.

The following diagram displays a more complex computed expression that strings together several functions, fields, field values, and operators to perform a calculation in a computed field:

**NOTE** For illustration purposes, the diagram outlines *some* of the functions, fields, field values, and operators.



The computed expression in the computed field determines if the item is a valid Change Order.

**IMPORTANT** Before you create a computed expression, MKS recommends that you review the following key considerations to understand acceptable syntax, operators, and constants.

## Key Considerations

### ■ General

- It is outside the scope of this guide to provide explicit instructions on how to write a computed expression. However, this guide does provide several examples of simple and complex computed expressions to illustrate some of the ways you can perform calculations in fields, charts, and reports.
- If you do not provide proper syntax for a computed expression, MKS Integrity displays a detailed error message when you save the computed expression.

---

- **Syntax**

- To specify field names, surround the field name with double quotation marks, for example, "Actual Cost".
- Spaces and special characters are ignored, except within quotations, for example, "Defect Count".
- Identifiers followed by "(" (a parenthesis) are considered functions. Identifiers not followed by a parenthesis are considered field names.
- Computed fields are valid as fields used in another computed field or computed expression; however, they may not recurse.

- **Arithmetic Operators**

Computed expressions support the following arithmetic operators:

- \* (multiplication)
- / (division)
- + (addition)
- - (subtraction)
- - (unary negation)
- Boolean-exp?true\_exp:false\_exp (conditional ternary)

- **Boolean Operators**

Computed expressions support the following boolean operators:

- == (equal)
- = (contains)
- != (not equal)
- <> (does not contain)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)
- and
- or

For boolean fields, two keywords are boolean constants: `true` and `false`. When a computed field is marked against a boolean field, the result of the computed expression must be a boolean value, for example:

```
expression boolean-operator expression
```

---

- **Empty Fields**

In MKS Integrity, a field without a value is referred to as *empty*. A computed expression that includes an empty field returns an empty value, except in aggregate expressions.

For example, to determine if the Project End Date date field is empty, type:

```
"Project End Date"-today() > 0
```

or

```
"Project End Date" - today() <=0? false:true
```

To specify an empty field value as zero, use the isEmpty function. For example:

```
isEmpty("QA Time", 0) + isEmpty("Development Time", 0)
```

allows you to add two time durations and retrieve a result, even if one of the values is empty.

- **Dates, Times, and Timezones**

- Dates and dates/times are considered separate date types in computed expressions, and cannot be assigned to one another.
- Displayed date fields do not change based on the time zone that a user is in; however, displayed date/time fields and time entries vary based on the time zone that a user is in.
- Computed expressions return dates/times in the MKS Integrity Client's time zone and perform calculations in the MKS Integrity Server's time zone, where appropriate.
- Computed expressions do not include a function for converting date/time information to date-only. This applies in all cases where an expression returns a timestamp (date plus time) value. Where date/time information is required, use the computed expression with a timestamp field.
- Date expressions accept all standardized date and time stamp formats.
- The following operations are supported with time stamps in computed expressions:

---

**IMPORTANT** You cannot mix dates and timestamps.

---

- timestamp + integer constant; integer constant + timestamp
- timestamp - integer constant
- timestamp("timestamp-constant")
- timestamp - timestamp
- timestamps can be subtracted or added to a constant integer, for example, now() +5 or now() - 5

Integer constants are treated in units of days; for example, timestamp + integer constant takes the specified time and adds a specified number of days. When subtracting two timestamps, the number of days between the two dates rounded to the nearest day is the result. Using other operators or a floating point results in an error.

---

- **User and Group Fields**

- You can compare single- and multi-valued user and group fields in computed expressions, for example, "Assigned User" = jriley, or "Assigned Group" = Development. By creating a computed expression that includes a user or group field comparison, you can incorporate the result of the computed field into an editability rule.
- You can also compare two user fields in a computed expression, for example, "Assigned User" = "Created By". A logical value or true or false is returned.
- You can also use the symbolic `me` user name to indicate the user retrieving the item.

---

# Computed Expression Function Classes

In a computed expression, you can use the following function classes to perform calculations:

- **Arithmetic Functions**

An *arithmetic function* can be applied in any computed expression type to perform basic arithmetic calculations. For example, adding two field values and rounding the resulting value to the nearest integer.

For a complete list of arithmetic functions, see “Arithmetic Functions” on page 166.

- **Date/Time Functions**

A *date/time function* is used to return or quantify date/time information. For example, the number of days an item spends in a specific state, or the earliest date/time recorded against an item.

For a complete list of date/time functions, see “Date/Time Functions” on page 167.

- **Aggregate Functions**

An *aggregate function* is used to perform aggregate calculations of item data, for example, using the `sum` function to add the field values in a list of items. Attempting to use an aggregate function for a normal expression results in an error.

For a complete list of aggregate functions, see “Aggregate Functions” on page 170.

- **Text Functions**

A *text function* is used to perform calculations with text and numeric fields to provide a text/numeric value. For example, using an item’s ID to create a computed field that calculates a unique identifier for an item, such as `REQ-00001234`. String expressions in text functions accept short and long text fields.

For a complete list of text functions, see “Text Functions” on page 172.

- **External Information Functions**

An *external information function* operates against a single item and extracts metric information from the item, for example, how many times the item has been in the `Submit` state. Attempting to use an external information function in an aggregate expression results in an error. For example, `SICPEntryCount()` is an invalid aggregate expression, but `sum(SICPEntryCount())` is valid because the argument to an aggregate function is a normal expression. `SICPEntryCount()` is part of the normal expression and is therefore valid.

For example, it is valid to use `DaysInState()` when accessing a single item, since it results in a single number; however, if you use it against a list of items, no results can be returned. To apply an external information function to each item in a list, embed it in an aggregate function, for example, `avg(DaysInState())`.

For a complete list of external information functions, see “External Information Functions” on page 173.

- **Item Information Functions**

An *item information function* is used to return basic information about an item type. For example, if an item type has a test management role of Test Case, a true or false value is returned.

For a complete list of item information functions, see “Item Information Functions” on page 175.

---

## ■ Test Functions

A *test function* is used to return information about test results. For example, return the numeric verdict type value for a given verdict ID.

For a complete list of test functions, see “Test Functions” on page 176.

## Key Considerations

- All fields return values specific to their field type; however, picklist fields return their numeric (non-textual) values. For example, `getFieldValue (567, Priority)` displays 1 as the field value for **Priority=High** for Defect 567.
- Node, segment, and shared item, and test types are used by the document model.
- Text strings in text calculations cannot exceed 1 kilobyte. Exceeding this limit may cause the MKS Integrity Server to fail.

## Arithmetic Functions

Name and Description	Return Value
<code>abs(numeric-expression)</code> Returns absolute value of numeric expression.	Integer or floating point (same as expression)
<code>sign(numeric-expression)</code> Returns -1 if expression evaluates to < 0, 0 if it evaluates to 0, and +1 if it evaluates to positive number.	Integer or floating point (same as expression)
<code>round(numeric-expression)</code> Returns expression rounded to nearest integer.	Integer
<code>truncate(numeric-expression)</code> Returns expression truncated to nearest integer.	Integer
<code>floor(numeric-expression)</code> Returns largest integer less than or equal to given numeric expression.	Integer
<code>ceil(numeric-expression)</code> Returns smallest integer greater than or equal to given numeric expression.	Integer
<code>mod(numeric-expression1, numeric-expression2)</code> Returns remainder of first expression divided by second expression. If either expression is not integer, then expression is: <code>(abs(e1) - floor(abs(e1)/abs(e2)) * abs(e2)) * sign(e1)</code>	Integer if both expressions integers; otherwise, floating point
<code>isEmpty(expression1, expression2)</code> First expression evaluated. If result not empty, it becomes return value for function. If empty, second expression evaluated and is return value for function. Both expressions must evaluate to same type. Expression can be numeric, time, or Boolean.	Type in expression
<code>emptyint()</code> Returns an empty value in an integer field.	none
<code>emptyfloat()</code> Returns an empty value in a floating point field.	none
<code>emptytime()</code> Returns an empty value in a date/time field.	none
<code>emptydate()</code> Returns an empty value in a date field.	none

Name and Description	Return Value
emptylogical() Returns an empty value in a logical field.	none
emptytext() Returns an empty value in a text field.	none
emptyuser() Returns an empty value in a user field.	none
emptygroup() Returns an empty value in a group field.	none
timestamp (quoted-string) Converts supplied string into timestamp constant. Using Java SimpleDateFormat, following formats are attempted in order until one succeeds: ■ Java DateTimeInstance (LONG, LONG) ■ Java DateTimeInstance (SHORT, SHORT) ■ MMM d, yyyy hh:mm a ■ MMM d, yyyy HH:mm ■ MMM d, yyyy HH:mm:ss	Timestamp
<b>Note:</b> Parsing done on the client using client's locale. Timestamp evaluated into timestamp constant during parsing. When expression displayed, displays in standard format for locale.	
now() Returns current time. <b>Note:</b> Current time is time when expression evaluated not parsed. To return number of days an item has existed, type: now() - "Created Date"	Timestamp
SelectionCount(field-name) Returns the number of entries selected in a multi-valued field on an item. If the field is empty, returns 0; if the field is not empty but is not multi-valued, returns 1. Not valid for attachment, relationship, or computed fields.	Integer

### Date/Time Functions

Name and Description	Return Value
DateDiff(date/time field1, date/time field2) Returns number of seconds between two date/time fields. If date/time field1 larger than date/time field2, result positive, otherwise negative. By including the dateFirstEntered function, you could calculate the number of seconds to close an Incident item, for example, DateDiff(dateFirstEntered("Closed"), "Created Date"). <b>Note:</b> To correctly calculate the expression, the <b>Include Time</b> option must be enabled in each specified date/time field.	Integer
DaysInState(state-name) Returns number of days item in specific state rounded to nearest day. Individual state times are added together in seconds; the resulting sum is rounded to days. To specify unspecified state, type "Unspecified", for example, DaysInState("Unspecified").	Integer
DaysInPhase(phase name, phase name) Returns number of days item in specific phase rounded to nearest day. Individual phase times are added together in seconds; the resulting sum is rounded to days.	Integer
DaysCurrentState() Returns number of days item in current state rounded to nearest day.	Integer
DayOfYear(date/timestamp/field) Returns the day of the year for the specified date.	Integer
dayOfWeek(date/timestamp/field) Returns the day of the week for the specified date. Sunday is 1, Monday is 2, and so on.	Integer

Name and Description	Return Value
<code>weekOfYear(date/timestamp/field)</code> Returns the week of the year for the specified date. <b>Note:</b> MKS Integrity does not support ISO-8601 for calendar weeks; therefore, the returned calendar week is always one number higher than expected. For example, for the year 2010 in MKS Integrity, calendar week 1 begins on January 1, 2010. In ISO-8601, calendar week 1 begins on January 4, 2010.	Integer
<code>monthOfYear(date/timestamp/field)</code> Returns the month of the year for the specified date.	Integer
<code>weeksDiff(dateA, dateB)</code> Returns the number of weeks between two specified dates. <b>Note:</b> This is the number of actual weeks, not the number of days divided by 7.	Integer
<code>monthsDiff(dateA, dateB)</code> Returns the number of months between two specified dates. <b>Note:</b> This is the number of actual months, not the number of days divided by 30.	Integer
<code>weekdaysDiff(dateA, dateB)</code> Returns the number of week days between two specified dates. Saturdays and Sundays are not included.	Integer
<code>DayOfWeekName(date/timestamp)</code> Returns day of week for the specified date or date and time, for example, Monday.	Text
<code>getDay(date/timestamp/field)</code> Returns the date for the specified date or date and time, for example, 23.	Integer
<code>getYear(date/timestamp/field)</code> Returns the date for the specified date or date and time, for example, 2009.	Integer
<code>MonthOfYearName(date/timestamp/field)</code> Returns month of the year for the specified date or date and time, for example, October.	Text
<code>getHour(timestamp/field)</code> Returns the hour for the specified time, for example, 2. <b>Note:</b> Hours are specified in 24-hour time. For example, midnight is 0 and 7pm is 19.	Integer
<code>getMinute(timestamp/field)</code> Returns the minute for the specified time, for example, 05.	Integer
<code>getSecond(timestamp/field)</code> Returns the second for the specified time, for example, 33.	Integer
<code>DaysCurrentPhase (phase-field-name)</code> Returns number of days item in current phase rounded to nearest day.	Integer
<code>dateFirstEntered(state-name)</code> Returns date specified state first entered.	Timestamp
<code>dateLastEntered(state-name)</code> Returns date specified state last entered.	Timestamp
<code>SecondsInState</code> Returns number of seconds item in specific state. If item in state multiple times, seconds for each occurrence added together. To specify unspecified state, type "Unspecified", for example, SecondsInState("Unspecified").	Integer
<code>SecondsInPhase</code> Returns number of seconds item in specific phase. If item in phase multiple times, seconds for each occurrence added together. To specify unspecified phase, type "Unspecified", for example, SecondsInPhase("Unspecified").	Integer
<code>SecondsCurrentState</code> Returns number of seconds item in current state.	Integer
<code>SecondsCurrentPhase</code> Returns number of seconds item in current phase.	Integer

Name and Description	Return Value
<pre>sumTimeEntry[(date("startdate"), date("enddate"))] or sumTimeEntry[(symbolicdate(), symbolicdate())]</pre> <p>Returns total time spent (rounded to hour) on item in optional timeframe.</p>	Integer
<pre>sumTimeEntryByUser(user[,user...] [,date("startdate"),date("enddate")]) or sumTimeEntryByUser(user[,user...] [,symbolicdate(),symbolicdate()])</pre> <p>Returns total time spent (rounded to hour) on item by specified users in optional timeframe.</p>	Integer
<pre>sumTimeEntryByGroup(group[,group...] [,date("startdate"),date("enddate")]) or sumTimeEntryByGroup(group[,group...] [,symbolicdate(),symbolicdate()])</pre> <p>Returns total time spent (rounded to hour) on item by specified groups in optional timeframe.</p>	Integer
<pre>sumTimeEntryByPhase(phaseField, phase[,date("startdate"),date("enddate")]) or sumTimeEntryByPhase(phaseField, phase[,symbolicdate(),symbolicdate()])</pre> <p>Returns total time spent (rounded to hour) on item while in phase for specified phase field.</p>	Integer
<pre>sumTimeEntryByPhaseByUser(phaseField, phase, user[,user...][,date("startdate"), date("enddate")]) or sumTimeEntryByPhaseByUser(phaseField, phase, user[,user...][,symbolicdate(),symbolicdate()])</pre> <p>Returns total time spent (rounded to hour) on item by specified users while in phase for specified phase field.</p>	Integer
<pre>sumTimeEntryByPhaseByGroup(phaseField, phase, group[,group...][,date("startdate"), date("enddate")]) or sumTimeEntryByPhaseByGroup(phaseField, phase, group[,group...][,symbolicdate(), symbolicdate()])</pre> <p>Returns total time spent (rounded to hour) on item by specified groups while in phase for specified phase field.</p>	Integer
<pre>sumTimeEntryByState(state[,date("startdate"), date("enddate")]) or sumTimeEntryByState(state[,symbolicdate(), symbolicdate()])</pre> <p>Returns total time spent (rounded to hour) on item while in specified state.</p> <p><b>Note:</b> MKS Integrity gathers time entries on a daily basis and assigns the time entry to the date (to midnight). MKS Integrity records state changes to the exact milli-second. Using time entries to determine how long an item has been in a particular state is only an approximation. Therefore, this function is only guaranteed correct if the item contains only one state transition for the entire day.</p> <p>For example, an item is posted, moves through several states (Investigate, In Development, Development Done), and is closed (Built) in a single day. Then, the user working on the item enters the total time spent on the item in a time entry. Because MKS Integrity cannot subdivide this time entry (hours spent on the item) into multiple pieces of information (hours spent in each state), the sumTimeEntryByState function assigns the time entry to a single state – the state that the item had on midnight on the day of the entry (Built).</p> <p>To specify unspecified state, type "Unspecified", for example, sumTimeEntryByState("Unspecified").</p>	Integer
<pre>sumTimeEntryByStateByUser(state, user[,date("startdate"),date("enddate")]) or sumTimeEntryByStateByUser(state (user[,symbolicdate(),symbolicdate()]))</pre> <p>Returns total time spent (rounded to hour) on item by specified users while in specified state.</p>	Integer
<pre>sumTimeEntryByStateByGroup(state, group[,date("startdate"),date("enddate")]) or sumTimeEntryByStateByGroup(state, group[,symbolicdate(),symbolicdate()])</pre> <p>Returns total time spent (rounded to hour) on item by specified groups while in specified state.</p>	Integer

Name and Description	Return Value
<pre>countTimeEntry(date("startdate"), date("enddate")) or countTimeEntry(symbolicdate(),symbolicdate()) </pre> <p>Returns number of time entries for item with duration greater than zero in optional timeframe.</p>	Integer
<pre>countTimeEntryByUser(user[,user...] [,date("startdate"),date("enddate")]) or countTimeEntryByUser(user[,user...] [,symbolicdate(),symbolicdate()]) </pre> <p>Returns number of time entries for item by specified users with duration greater than zero in optional timeframe.</p>	Integer
<pre>countTimeEntryByGroup(group[,group...] [,date("startdate"),date("enddate")]) or countTimeEntryByGroup(group[,group...] [,symbolicdate(),symbolicdate()]) </pre> <p>Returns number of time entries for item by specified groups with duration greater than zero in optional timeframe.</p>	Integer
<pre>firstTimeEntryDate() </pre> <p>Returns earliest date time recorded against item.</p>	Date
<pre>lastTimeEntryDate() </pre> <p>Returns latest date time recorded against item.</p>	Date

## Aggregate Functions

Name and Description	Return Value
<pre>sum(numeric-expression) </pre> <p>Adds expressions calculated against each item that the aggregation running against.</p>	Integer or floating point, same as expression
<pre>avg(numeric-expression) </pre> <p>Adds expressions calculated against each item that the aggregation running against, then divide by number of non-empty entries.</p>	Floating point
<pre>min(numeric-expression or timestamp) </pre> <p>Finds smallest of expressions from each item that the aggregation running against. If all expressions empty, result empty.</p>	Integer, float point, or timestamp same as expression
<pre>max(numeric-expression or timestamp) </pre> <p>Finds largest of the expressions from each item that the aggregation is running against. If all expressions are empty, the result is empty.</p>	Integer, floating point, or timestamp same as expression
<pre>stddev(numeric-expression) </pre> <p>Returns the variation above or below the mean (average) value. For example, you can calculate the standard deviation of recorded time estimates. If all expressions are empty, the result is ignored. If there are less than two non-empty values, the result is zero. You can use the <code>stddev</code> aggregate function in the <code>Aggregate</code> or <code>Query</code> external information functions, or in a report that uses the <code>groupcompute</code> tag.</p>	Floating point
<pre>variance(numeric-expression) </pre> <p>Returns the difference between the expected and actual result. For example, you can calculate the difference between a budget and actual expenditure. If all expressions are empty, the result is ignored. If there are less than two non-empty values, the result is zero. You can use the <code>variance</code> aggregate function in the <code>Aggregate</code> or <code>Query</code> external information functions, or in a report that uses the <code>groupcompute</code> tag.</p>	Floating point
<pre>count() </pre> <p>Returns number of items that the aggregation is running against.</p>	Integer

Name and Description	Return Value
<pre>sumTimeEntrySecs [(date("startdate"), date("enddate"))] or sumTimeEntrySecs [(symbolicdate(), symbolicdate())]</pre> <p>Returns the total time spent (in seconds) on an item in an optional timeframe.</p>	Integer
<pre>sumTimeEntrySecsByUser (user[, user...][, date("startdate"), date("enddate")]) or sumTimeEntrySecsByUser (user[, user...][, symbolicdate(), symbolicdate()])</pre> <p>Returns the total time spent (in seconds) on an item by specified users in an optional timeframe.</p>	Integer
<pre>sumTimeEntrySecsByGroup (group[, group...][, date("startdate"), date("enddate")]) or sumTimeEntrySecsByGroup (group[, group...][, symbolicdate(), symbolicdate()])</pre> <p>Returns the total time spent (in seconds) on an item by specified groups in an optional timeframe.</p>	Integer
<pre>sumTimeEntrySecsByState (state[, date("startdate"), date("enddate")]) or sumTimeEntrySecsByState (state[, symbolicdate(), symbolicdate()])</pre> <p>Returns the total time spent (in seconds) on an item while in the specified state.</p>	Integer
<pre>sumTimeEntrySecsByStateByUser (state, user[, date("startdate"), date("enddate")]) or sumTimeEntrySecsByStateByUser (state (user[, symbolicdate(), symbolicdate()]))</pre> <p>Returns the total time spent (in seconds) on an item by specified users while in the specified state.</p>	Integer
<pre>sumTimeEntrySecsByStateByGroup (state, group[, date("startdate"), date("enddate")]) or sumTimeEntrySecsByStateByGroup (state, group[, symbolicdate(), symbolicdate()])</pre> <p>Returns the total time spent (in seconds) on an item by specified groups while in the specified state.</p>	Integer
<pre>sumTimeEntrySecsByPhase (phaseField, phase[, date("startdate"), date("enddate")]) or sumTimeEntrySecsByPhase (phaseField, phase[, symbolicdate(), symbolicdate()])</pre> <p>Returns the total time spent (in seconds) on an item while in phase for the specified phase.</p>	Integer
<pre>sumTimeEntrySecsByPhaseByUser (phaseField, phase, user[, user...][, date("startdate"), date("enddate")]) or sumTimeEntrySecsByPhaseByUser (phaseField, phase, user[, user...][, symbolicdate(), symbolicdate()])</pre> <p>Returns the total time spent (in seconds) on an item by specified users while in phase for the specified phase.</p>	Integer

Name and Description	Return Value
<pre>sumTimeEntrySecsByPhaseByGroup (phaseField, phase, group[,group...][,date("startdate"), date("enddate")]) or sumTimeEntrySecsByPhaseByGroup (phaseField, phase, group[,group...][,symbolicdate(),symbolicdate()]) Returns the total time spent (in seconds) on an item by specified groups while in phase for the specified phase.</pre>	Integer
<pre>LastResultAgg() Returns the internal ID of the most recent test result record for a group of items. This function must be used with the Aggregate, AggregateByTree, or Query functions. For example, AggregateByTree (LastResultAgg()). This function must be used with the TestVerdictName function to convert the internal ID to a display string. For example, TestVerdictName(LastResultAgg()) .</pre>	Integer

### Text Functions

Name and Description	Return Value
<pre>Text(string) Creates text constant.</pre> <p><b>Note:</b> If string does not contain special characters, quotes not required. For example, <code>text ("REQ")</code> and <code>text (REQ)</code> identified as same string.</p>	Text
<pre>Length(string-expression) Returns length of string.</pre>	Integer
<pre>Upper(string-expression) Converts string into uppercase.</pre>	Text
<pre>Lower(string-expression) Converts string into lowercase.</pre>	Text
<pre>Concat(string-expression, string-expression[, ...]) Concatenates two or more strings.</pre>	Text
<pre>Substring(string-expression, start, count) Takes substring of string. Substring starts at first character and goes for count characters. start offset is origin 1. start and count must be positive integer constants.  Following conditions return empty string: ■ start less than or equal zero. ■ start greater than length of string being processed. ■ count less than zero. ■ If start plus length greater than length of string being processed, then return from start to end of string; no blank padding.</pre>	Text
<pre>Locate(string-expression, string-expression) Searches for first string in second string.  To add startsWith, type Locate(x, y) == 1.  Value offset into second string first string found, where if second string starts with first, value one (1). If not found, returns zero (0).</pre>	Integer
<pre>Trim(string-expression) Removes leading and trailing spaces from string.</pre>	Text
<pre>LTrim(string-expression) Removes leading spaces from string.</pre>	Text
<pre>RTrim(string-expression) Removes trailing spaces from string.</pre>	Text
<pre>ToText(number-expression) Converts number to string.</pre>	Text

Name and Description	Return Value
ToTextZeroPad(number-expression, size) Converts number to string with zero padding. size must be positive integer constant. If formatted size of number greater than or equal to specified size, it returns unchanged; otherwise, padded on left with the zero (0) character. If number negative, negative sign (-) first character.	Text
CurrentUser() Returns user name of user currently viewing the item. The user name can then be used in other computed functions.	Text

### External Information Functions

Name and Description	Return Value
SICPCount([cpstate, ...]) Returns number of change packages associated with item. Specify one or more of following change package states to include in count: <ul style="list-style-type: none"><li>■ Closed</li><li>■ Open</li><li>■ Submitted</li><li>■ Accepted</li><li>■ Rejected</li><li>■ Discarded</li><li>■ CommitFailed</li></ul> If you do not specify any arguments, count of all change packages returned.	Integer
SICPEntryCount([cpentrytype, ...]) Returns number of change package entries associated with item. Specify one or more of following change package entry types to include in count: <ul style="list-style-type: none"><li>■ Update</li><li>■ Add</li><li>■ Drop</li><li>■ Lock</li><li>■ RenameFrom</li><li>■ RenameTo</li><li>■ UpdateRevision</li><li>■ UpdateArchive</li><li>■ AddFromArchive</li><li>■ MoveMemberFrom</li><li>■ MoveMemberTo</li><li>■ CreateSubproject</li><li>■ AddSubproject</li><li>■ AddSharedSubproject</li><li>■ DropSubproject</li><li>■ ConfigureSubprojectFrom</li><li>■ ConfigureSubprojectTo</li><li>■ MoveSubprojectFrom</li><li>■ MoveSubprojectTo</li></ul> If you do not specify any arguments, count of all unique change package entry operations returned, for example, rename operation only counted as one entry rather than separate entries for RenameFrom and RenameTo operations.	Integer
SICPBytesAdded() Returns sum of bytes added by each change package entry for all change packages associated with item. Returns 0 for binary files. Specified arguments must be text constant suffix matches for the members in the change package. For example, SICPBytesAdded(".java", ".c").	Integer

Name and Description	Return Value
SICPBytesDeleted() Returns sum of bytes deleted by each change package entry for all change packages associated with item. Returns 0 for binary files. Specified arguments must be text constant suffix matches for the members in the change package. For example, SICPBytesDeleted(".java", ".c").	Integer
SIDaysCPOpen() Returns total number of days changes packages associated with item open. If change package currently open, current time used to calculate number of days. If function used based on date and time in past, count based on specified time. Any change packages created after specified time not counted. Number of days open for any change packages closed after specified time calculated using specified time as close time.	Integer
SICPLinesAdded() Returns sum lines added by each change package entry for all change packages associated with item. Returns 0 for binary files. Specified arguments must be text constant suffix matches for the members in the change package. For example, SICPLinesAdded(".java", ".c").	Integer
SICPLinesDeleted() Returns lines deleted by each change package entry for all change packages associated with item. Returns 0 for binary files. Specified arguments must be text constant suffix matches for the members in the change package. For example, SICPLinesDeleted(".java", ".c").	Integer
SIMetric(siproject-field-name, metric name) Returns calculated value of metric for configuration management project.	Integer
SIMetricCount(siproject-field-name, metric name) Returns number of items that make up metric value for configuration management project.	Integer
NumberOfEntriesToState(state-name) Returns number of times specified state entered. To specify unspecified state, type "Unspecified", for example, NumberOfEntriesToState("Unspecified").	Integer
numberOfHistoryEntries() Returns number of deltas associated with item. Means number of times item edited.	Integer
numberOfModifications(field-name) Returns number of deltas specified field changed in.	Integer
HistoricFieldValue(field-name, timestamp-constant) Returns value of specified field at specific point in history. <b>Warning:</b> Operation can take long time to complete.	Type of specified field name
RelCount(relationship-field) Returns number of related items through specified relationship field. Function equivalent to: aggregate(relationship-field, count()).	Integer
RelExists(relationship-field) Returns whether relationships exist through specified relationship field.	Boolean
Aggregate(relationship-field, ByDocument(recurseInclude, recurseReference) [, ...], aggregate-expression) Aggregate operates against a single item, using that single item to find multiple items that it then runs an aggregate expression against to product a single value. <b>Note:</b> Aggregate is <i>not</i> an aggregate function. At least one relationship field must be specified. A set of items is made by following the first relationship field. Then that set is augmented by all items that the set points to through the next relationship field, and onward for each specified relationship field. The ByDocument function treats items found through the relationship field as documents, find all their nodes, and recurses into included and referenced documents.	Type of aggregate expression

Name and Description	Return Value
<pre>Query(query-name-string[, correlation-field[, ...]], aggregate-expression)</pre> <p>Operates against single item using that single item to find multiple items it aggregates into for single value. query-name-string is name of query as quoted string in syntax "username:queryname". If no correlation-fields given, value of current item not used, and result of expression is constant. Aggregation expression runs against all items that satisfy query. Otherwise, query modified to require each specified correlation-field to match source item. Resulting list of items has aggregate-expression run against it and returned.</p>	Integer or float, depending on aggregation-expression
<pre>Query(query-name-string[, source-correlation-field, target-correlation-field[, ...]], aggregate-expression)</pre> <p>More generalized form of the previous Query function; however, if source-correlation-field is the same as target-correlation-field, use the previous Query function. This Query function takes list of fields that match between source and target items. Useful when you want to match between master item that has field set equal to values on all related items, in particular, <b>Project</b> field.</p>	Integer or floating point, depending on aggregation-expression; see Query function
<pre>QueryCorrelated(query-name-string[, source-correlation-field, target-correlation-field[, ...]], aggregate-expression)</pre> <p>To express Query as QueryCorrelated function, specify aggregation-field from Query as both source-correlation-field and target-correlation-field.</p> <p>Allows you to specify matching fields on both sides, which is useful for item backed pick list (IBPL) fields. For example, if you have an item type that defines a set of something, like applications, the IBPL links each member of application back to application while a query backed relationship (QBR) field is used for linking between each application to each of its members. You can use the QueryCorrelated function on application using item ID from Application item correlated with IBPL representing Application on Defect that is a member of it, for example:</p> <pre>QueryCorrelated("All Defects", ID, "Application IBPL", count())</pre>	Integer or floating point, depending on aggregation-expression; see Query function

### Item Information Functions

Name and Description	Return Value
<pre>getFieldValue(item#, field)</pre> <p>Returns a field value in a specific item ID. For example, getFieldvalue (234, "Found in Build") returns a <b>Found in Build</b> field value of 7415 for Defect 234.</p>	Depends on the field type
<pre>IsState(statename[, ...])</pre> <p>Returns true if item's state any of states specified in its argument list.</p> <p>To specify unspecified state, type "Unspecified", for example, IsState ("Unspecified").</p>	Boolean
<pre>IsType() (typename[, ...])</pre> <p>Returns true if item's type any of types specified in its argument list.</p>	Boolean
<pre>IsNode()</pre> <p>Returns true if item any of type node.</p>	Boolean
<pre>IsSegment()</pre> <p>Returns true if item any of type segment.</p>	Boolean
<pre>IsSharedItem()</pre> <p>Returns true if item any of type shared item.</p>	Boolean
<pre>IsContent()</pre> <p>Returns true if item any of types content.</p>	Boolean
<pre>IsSubsegment()</pre> <p>Returns true if item any of type subsegment.</p>	Boolean
<pre>IsMeaningful()</pre> <p>Returns true if item any of type node or segment is meaningful.</p>	Boolean
<pre>IsNonMeaningful()</pre> <p>Returns true if item any of type node or segment is non meaningful.</p>	Boolean
<pre>IsGroupDocument()</pre> <p>Returns true if item any of type segment what has the Group Document flag set.</p>	Boolean
<pre>IsTestCase()</pre> <p>Returns true if item type has a test management role of Test Case.</p>	Boolean

Name and Description	Return Value
<code>IsTestSession()</code> Returns true if item type has a test management role of Test Session.	Boolean
<code>IsTestStep()</code> Returns true if item type has a test management role of Test Step.	Boolean
<code>IsTestSuite()</code> Returns true if item type has a test management role of Test Suite.	Boolean

### Test Functions

Name and Description	Return Value
<code>TestVerdict(test-verdict-name)</code> A test verdict name constant that can be used in a comparison. For example, <code>LastResult() == TestVerdict(Passed)</code> .	Text
<code>TestVerdictByType(verdict-type-name)</code> A test verdict type constant corresponding to the pass, fail, or other.	Text
<code>TestVerdictName(id)</code> Returns the display name of the verdict corresponding to the ID. For example, <code>TestVerdictName(TestVerdict(Passed))</code> would return Passed; <code>TestVerdictName&gt;LastResult()</code> would return the last test result as a string.	Text
<code>TestVerdictTypeName(verdict-type-id)</code> Returns the name of the test verdict type. For example, <code>TestVerdictTypeName(TestVerdictByType(pass))</code> would return Pass	Text
<code>VerdictToType(verdict-id-expression)</code> Returns the numeric verdict type value for the given verdict ID. Can be applied to <code>LastResult()</code> to check the verdict type of a result. For example, <code>VerdictToType&gt;LastResult() == TestVerdictByType(pass)</code> would check if a result was a pass verdict type.	Integer
<code>ResultCount(test-verdict-name)</code> Returns the number of test results with a specific verdict. For example, <code>ResultCount(Passed)</code> would return the number of successful test results. This function is intended to be used with test session type items.	Integer
<code>ResultCountByType(verdict-type-name)</code> Returns the number of test results with a specific verdict type (pass, fail, other). For example, if you have two test verdicts with a verdict type of fail, the function <code>ResultCountByType(Fail)</code> would return the number of test results with either verdict. This function is intended to be used with test session type items.	Integer
<code>ResultCount(Relationship-field, test-verdict-name)</code> Returns the number of test results with a specific verdict in related test session type items.	Integer
<code>ResultCountByType(Relationship-field, verdict-type-name)</code> Returns the number of test results with a specific verdict type (pass, fail, other) in related test session items.	Integer
<code>LastResult()</code> Returns the internal ID of the most recent test result for the current item. This function must be used with the <code>TestVerdictName</code> function to convert the internal ID to a display string. For example, <code>TestVerdictName&gt;LastResult()</code> .	Integer

# Creating a Computed Expression in a Report

If your administrator enables computed expressions in report recipes, you can create computed expressions in a report to perform calculations on item data. This is particularly useful when you do not have permission or the ability to create a computed field in an item type that performs a similar calculation.

## **Example: Project Overview**

To provide an overview of project status, the report displays an overview of all projects, providing effort, budget, and project health information. Computed expressions in the report calculate effort and budget variances.

For example, in the **Report Wizard**, the following computed expressions display in the **Item Fields** panel:

The screenshot shows the Report Wizard interface with two main sections: 'Item Fields' and 'Computed Fields'.  
In the 'Item Fields' section, there is a list of fields: ID, Product, Summary, Project, and Project Phase (for Project, Library).  
In the 'Computed Fields' section, there are two entries:

Name	Expression	Display Pattern
Effort Variance	"Estimated Effort" == 0 ? 0.0 : ((Actual Effort)/8) - "Estimated Effort"	#%
Budget Variance	IsEmpty("Estimated Budget" - Actual Budget, 0.0)	\$#,###

With buttons for 'Add', 'Edit', and 'Remove' on the right.

Summary of computed expressions:

- `"Estimated Effort" == 0 ? 0.0 : ((Actual Effort)/8) - "Estimated Effort" / "Estimated Effort"` - calculates the variance between the **Actual Effort** and **Estimated Effort** fields.
- `IsEmpty("Estimated Budget" - Actual Budget, 0.0)` - calculates the variance between the **Estimated Budget** and **Actual Budget** fields.

The generated report displays the following data:

## Project Overview (All)

Overview of all projects.

Sep 28, 2010



ID	Product	Summary	Project	Project Phase	Actual Effort	Estimated Effort	Effort Variance	Actual Budget	Estimated Budget	Budget Variance	Health Value
104	Tempus Classic - (98)	Project for Release 1	/Embedded/Projects/Release1	Active	0 hrs	4,900 days	-100%	\$375,000	\$350,000	-\$25,000	1
105	Integrated Patient Monitor and Display - (101)	Project for Release 1	/Medical/Projects/Release1	Active	0 hrs	7,250 days	-100%	\$1,375,000	\$1,200,000	-\$175,000	-1
106	Adaptive Cruise Control System - (103)	Project for Release 1	/Automotive/Projects/Release1	Active	0 hrs	27,500 days	-100%	\$3,154,000	\$3,300,750	\$146,750	-1

# Creating a Computed Expression in a Chart

You can create computed expressions in charts to perform calculations on item data. This is particularly useful when you do not have permission or the ability to create a computed field in an item type that performs a similar calculation.

## **Example: Cost Summary of Active Projects**

In a distribution chart, use aggregate functions to calculate a cost summary of active projects for each product.

For example, in the **Chart Properties** dialog box, the following aggregate expressions display in the **Aggregation** panel:

Aggregation Expression		Expression Label	Display Pattern	Axis Label	
count()	# of Projects	#,###		Count	<a href="#">Add</a>
max("Product Health Value")	Health				<a href="#">Edit</a>
sum("Total Estimated Budget")	Total Estimated Cost	\$#,##0.00	\$		<a href="#">Move Up</a>
sum("Total Actual Budget")	Total Actual Cost	\$#,##0.00	\$		<a href="#">Move Down</a>
sum("Total Estimated Budget")-sum("Total Actual Budget")	Cost Variance	\$#,##0.00	\$		<a href="#">Remove</a>

Summary of aggregate expressions:

- `count()` - displays the number of active projects for each product.
- `max("Product Health Value")` - displays the **Health** of each product by calculating the maximum value of the **Product Health Value** computed field in each active project.
- `sum("Total Estimated Budget")` - displays the **Total Estimated Cost** of a product by calculating the sum of the **Total Estimated Budget** field in each active project.
- `sum("Total Actual Budget")` - displays the **Total Actual Cost** of a product by calculating the sum of the **Total Actual Budget** field in each active project.
- `sum("Total Estimated Budget")-sum("Total Actual Budget")` - displays the **Cost Variance** for a product by calculating the difference between the **Total Estimated Budget** field and **Total Actual Budget** field in each active project.

The generated chart displays the following data:

Product	# of Projects	Health	Total Estimated Cost	Total Actual Cost	Cost Variance
unspecified	0				
Adaptive Cruise Control System - (103)	1	●	\$3,300,750.00	\$3,154,000.00	● \$146,750.00
Cardiac Device - (102)	0				
Integrated Patient Monitor and Display - (101)	1	●	\$1,200,000.00	\$1,375,000.00	● -\$175,000.00
Tempus Classic - (98)	3	●	\$1,045,000.00	\$1,190,000.00	● -\$145,000.00
Tempus Modern - (100)	0				
Tempus Rugged - (99)	0				

## PART 7

# Working With Documents

# What is a Document?

A document is a collection of related MKS Integrity items. The **Document** view is the interface you use to view and modify documents and content in a cohesive tree structure. With the applicable permissions, you can manage all aspects of document and content creation and modification, including creating relationships between content items across and between domains, called *trace relationships*.

MKS Integrity uses specific terms to describe the components of a root document. Specifically, *segment* is a root document, the container for all content; *node* is any content within that root document including individual content items and subsegments, and *shared item* is the shared content.

Documents are typically used in the requirements management and test management domains, although numerous other domains of analysis are supported by the MKS Integrity document model. For example, a requirement item could be verified by a test case and that requirement and test case pair could be connected to a larger test suite document using a relationship that collectively verifies a major software change.

Test management and requirements management domains are part of ALM. For detailed document concepts and use cases, see the documentation included with the ALM solution download, available on the MKS Customer Community (<http://www.mks.com/community>).

## Understanding Document Structure

This section discusses components used in the Document view and throughout MKS Integrity specifically in reference to document and content items.

Similar to other items, relationships are the glue that ties document and content items together and allows for reuse and persistence of artifacts over time. To learn about the relationships inherent in the document model, see “Relationships in the Document Model” on page 183.

MKS Integrity Term	Architectural Term	Definition
Document (root)	Segment (root)	<ul style="list-style-type: none"><li>■ Container for nodes that reference shared items or other documents.</li><li>■ A root document is also considered a segment.</li><li>■ Represented by the top-level node in the Outline panel of the Document view.</li><li>■ May be related to the item backing the project via the <b>Documents</b> relationship. The subsegment refers to the node that references a segment root.</li></ul>
Document	Segment/Subsegment	<p>Segments:</p> <ul style="list-style-type: none"><li>■ are documents or subdocuments that contain content.</li><li>■ can include or reference other segments.</li></ul> <p>Subsegments:</p> <ul style="list-style-type: none"><li>■ are segments that are a part of another document. If they are independent of another document they are considered segments.</li><li>■ are included or inserted into a document which creates a <b>References/Referenced By</b> relationship between nodes/segment respectively.</li><li>■ can be nested within segments or other subsegments.</li></ul>

MKS Integrity Term	Architectural Term	Definition
Content	Node	<p>Content:</p> <ul style="list-style-type: none"> <li>■ is the term used in MKS Integrity for a node/shared item pair; the node references the shared item.</li> <li>■ can be meaningful, for example, a requirement, test case, segment or subsegment. It can also be non-meaningful; for example a heading or a comment. These criteria are defined by your system administrator on the <b>Shared Category</b> field.</li> </ul> <p>Nodes:</p> <ul style="list-style-type: none"> <li>■ are references to individual content items (can also be subsegments containing their own nodes) that belong to a document and contain document-specific metadata.</li> </ul>
Contents	n/a	All node/shared item pairs that are contained within a document.
Shared Item	n/a	<p>MKS Integrity items that represent shared content in your document model. Transparent during the creation or modification of content.</p> <p>Shared Items:</p> <ul style="list-style-type: none"> <li>■ are referenced by nodes.</li> <li>■ track modifications to content and reflect them in the node.</li> </ul>
Section	n/a	A grouped hierarchy of nodes in a document, for example: section 1.4 and its children, 1.4.1, 1.4.2 or section 3.4.2 and its children 3.4.2.1, 3.4.2.2.
Reference mode	n/a	Setting available on a node within a document that controls how the system behaves when content is reused and modified. Options are: Author, Reuse, Share.
Category	n/a	<p>Pick field exposed on a node or segment type which allows for the categorization of content items, for example: <b>Heading</b>, <b>Comment</b>, <b>Requirement</b>, <b>Test Case</b>.</p> <p>You can select a segment or node type from the <b>Category</b> pick list value when you create or modify a segment or a node.</p> <p>Available options are defined by your administrator on the <b>Shared Category</b> field.</p>
Group document	n/a	A read-only document used for grouping content. You cannot create or edit meaningful content in a group document.

## Reference Modes

All content within a document in MKS Integrity has specific permissions and access levels associated with it. Reference modes dictate how the system behaves when content is created and modified.

- **Author**

Author reference mode controls the evolution of content. Items with an author reference mode can be created or modified by anyone with the appropriate permissions within the project. Other references to the shared item will continue to refer to the old shared item.

- **Reuse**

Reuse reference mode allows you to view a fixed revision of the content. The shared item will branch if you attempt a significant edit. Significant edits are modifications you make to the significant fields on a node that are defined by your administrator. A branch is a copy of the original document. You can also branch content if you have the correct permissions. To learn about branching documents, see “Branching Documents” on page 212.

You can change a reference mode from share to reuse and from reuse to share.

- **Share**

Share reference mode allows you to view the current version of the content. You can watch the changes to content as content with author reference mode changes, but you cannot edit the

---

content directly. The shared item is uneditable and you participate in any changes made by the node from which you are sharing.

If the share relationship is changed through branching, the reference mode changes to Reuse to reflect the new relationship. If the share relationship is removed, the reference mode also changes to reuse.

You can change a reference mode from share to reuse and from reuse to share. However, an Author reference mode cannot be changed.

## Relationships in the Document Model

Document and content item relationships are specific to a document model environment often used in requirements management and test management domains. These relationships dictate the behavior between documents and nodes, nodes and nodes, and also between nodes and their shared items.

Document and/or content items that fulfil the relationships explained below can be viewed in the Item Details view for a content or document item.

---

**NOTE** The structural and couplet relationships are built in relationships in MKS Integrity.

---

The include form of nesting documents allows you to manage content as a single document. Dividing content into multiple documents may simply be for convenience reasons; for example, to partition work within the team, or to facilitate higher performance within the system.

### **Structural**

The structural relationship is used to construct the tree hierarchy or document structure. Each parent in the hierarchy can have one or more children. Each child can have only one parent.

The main structural relationship between the segments and nodes is called **Contains/Contained By**.

### **Couplet**

The couplet relationship is used to link the node to the shared item or the node to the segment. The nature of the couplet relationship is mediated by the **Reference Mode** pick list on the node. Each shared item or segment can be referenced by multiple nodes, while each node can reference only one shared item or segment.

The couplet relationship is called **References/Referenced By**.

### **Trace**

You can create trace relationships between two items in the document model repository; for example, between a requirement and a test case, or between a higher-level requirement and a lower-level requirement. Trace relationships are defined via field pairs and are presented to the user in domain-specific language, for example, test and requirements.

Although the amount of types of trace relationships is not constrained, generally there is one relationship between each type of object in the system, for example: high level requirements, marketing requirements, test cases, components.

Trace relationships are often used in the context of the Application Lifecycle Management (ALM) template. To learn more about trace relationships in the ALM solution, see the documentation included with the ALM solution download, available on the MKS Customer Community (<http://www.mks.com/community>).

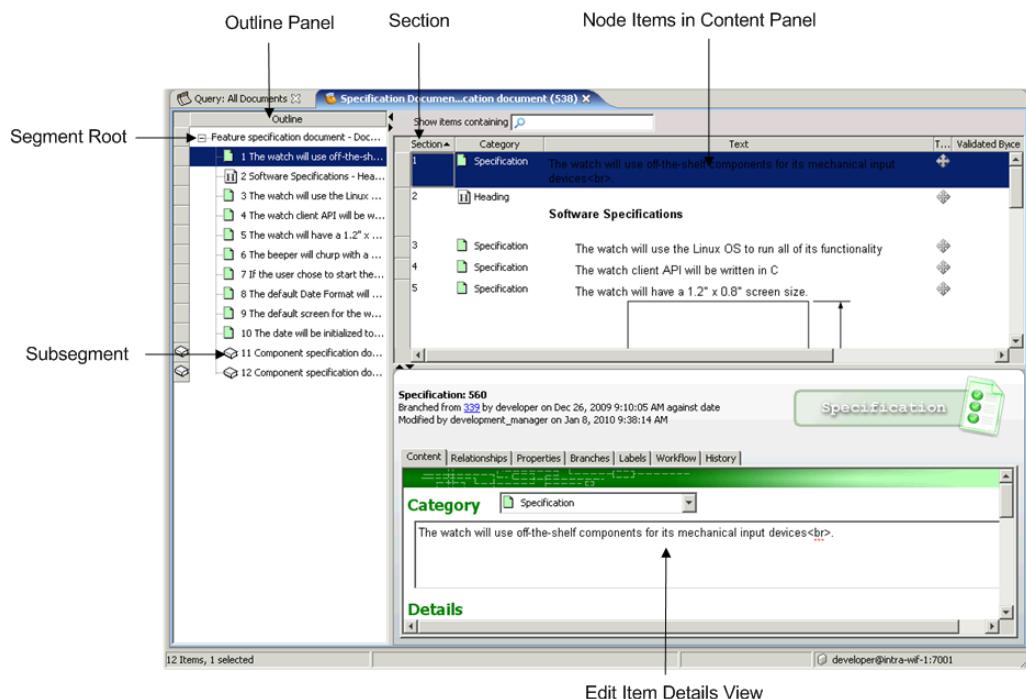
# Document View

The **Document** view allows you to manage all aspects of document and content creation and modification from one central location.

The Document view displays when you view or modify existing documents or create or new ones. To learn how to find and view existing documents, see “[Finding and Viewing Documents](#)” on page 203.

The **Document** and the **Content** menus allow you to create and modify documents and content. The **Document** menu operates on the document as a whole, and the **Content** menu operates on selected content within a document. You can have separate **Document** views displayed for different root document items.

To learn how to customize MKS Integrity to display the Document and Content menus, see your *MKS Integrity 2009 Getting Started Guide*.



*Document view with Item Details (Edit mode)*

To understand the terms and concepts inherent in the document and content tree structure, see “[Understanding Document Structure](#)” on page 181.

Some nodes on the Outline panel may be expandable and some may not be. Nodes can represent editable or uneditable subsegments or individual content items depending on how the content was created.

The connection between document items is based on relationship fields called traces and is entirely configurable. To learn more about trace relationships, see “[Managing Trace Relationships](#)” on page 215.

To learn more about trace propagation, see “[Propagating Traces to Branched Documents](#)” on page 218.

---

# Configuring the Document View

If you have the applicable permissions, you can configure the **Document** view to display the way you want to work with documents and content.

To change display attributes associated with the Document view, click **View > Options**. The **Options** dialog box displays with the **General** tab selected.

## **View Layout**

The **Outline** panel displays the document relationship hierarchy of the document or content you used to launch the view.

To learn more about document model relationships, see “Relationships in the Document Model” on page 183.

If it is not already enabled, to view the Outline panel in the Document view, click **View > Options** and enable the **Show Outline** option.

In the **General** tab, under **View Layout**, you can also define how the Outline panel and the Content panel (on the right in the Document view) behave when an item is selected:

- **Link with detail**

Each time an individual item is selected in the Outline panel, the same item is selected in the Content panel.

- **Link with outline**

Each time an individual item is selected in the Content panel, the same item is selected in the Outline panel.

- **Substitute parameters**

If an item has parameters specified in a text field and the field is set up to support parameter substitution, select this option to have the parameter replaced with the correct parameter value.

MKS Integrity searches a hierarchy of items that are related to the current item in order to determine what parameter value to substitute. If you change a parameter value in the document view, you must refresh the view in order to see the impact of the change on related document items.

## **Inline Editing**

The inline editing option expands the row to the height of the data contained in the fields, allowing you to edit fields in the Document view. In the **General** tab, under **Inline Editing**, click **Off** or **On** to toggle inline editing in the Document view.

## **Viewing or Editing Panel**

You can configure the Item Details view to be hidden, docked and editable, or docked and uneditable from the Document view by selecting **View > Options** and by choosing the applicable **View or editing panel** option from the **General** tab:

- **Off**
- **Show item edit view**
- **Show item detail view**

---

To edit a content item inline in a document, click the field you want to edit. If the field is a pick list, the list options display. Your administrator defines the available options in this list.

---

**NOTE** You cannot inline edit an item if a relevance rule is set that prevents it. Contact your type administrator for more information.

---

### **Tooltip Display**

By default, long field values in tooltips may be truncated. You can display the full value by enabling **Show full field value in tooltip**.

### **Operations Mode for Working in Documents**

When working within the Document view, you have additional options for managing actions such as copy, cut, paste, move, and drag. Under **File > Options > General** tab, the **Operations Mode** option can be set to **Default**, **Content**, or **Section**:

- The **Default** option maintains the previous behavior of the Document view. Selections on the left (outline) move both parent and children. Selections on the right move the selected item only, without its children.
- The **Content** option allows you to work on both sides of the view, moving or highlighting items without their children. The **Content** option does not preserve document structure.
- The **Section** option allows both sides of the view to have the same behavior. Actions on highlighted items select and act on the items and their children. The **Section** option preserves document structure.

---

**NOTE** When working with multiple open Document views, you can apply different settings to each view and those settings persist as long as the views are open. When you switch focus from one view to another, the settings change accordingly. If you open new Document views, the last modified settings are applied.

---

### **Colors**

You can define the cell, row, and font colors of mandatory fields and messages, row edit indicators, and levels by selecting **View > Options** and selecting the **Colors** tab.

### **Document Messages**

You can define whether or not to display pop-up messages when working in a root segment by selecting **View > Options** and choosing the **Messages** tab. In addition to defining the fields that display in the message, you can define the color of the messages by clicking the **Colors** tab.

The following types of messages can be configured:

Message Type	Displays/Indicates
Show the mandatory field information pop up	The message that displays for outstanding mandatory fields. The default message is: The field {FieldName} is mandatory and requires a value before it can be saved. You can create a custom message to display more specific information, for example, The field {FieldName} is a mandatory field on {Type} : {ID}. Click the <b>Fields</b> button to select and insert the fields in the message text.
Show a message when entering a subdocument	This message along with the associated document icon and tooltip in the row header allow you to easily identify what subdocument you are in. You can create a custom message to display more specific information, for example, You are now working in {Type} : {ID}. Click the <b>Fields</b> button to select and insert the fields in the message text.

Message Type	Displays/Indicates
Show confirmation message before saving a modified item	Message indicating that the document item you modified was saved.
Show the not editable field information popup	Message that displays when attempting to edit a non-editable field.
Show paste warning with filter applied	Message that displays if you paste content in the content pane that does not match the current text filter, warning you that the pasted content will not display.

### Outline and Title Attributes

You can configure how documents are represented in the Outline panel by selecting **View > Options > Titles**, and defining the format for the following:

- **Outline column format**
- **Title format**
- **Tab title format**

Enter values in the field surrounded by {} brackets.

Instead of manually entering in the field names, fields can be selected from the list and inserted by clicking the **Fields** button.

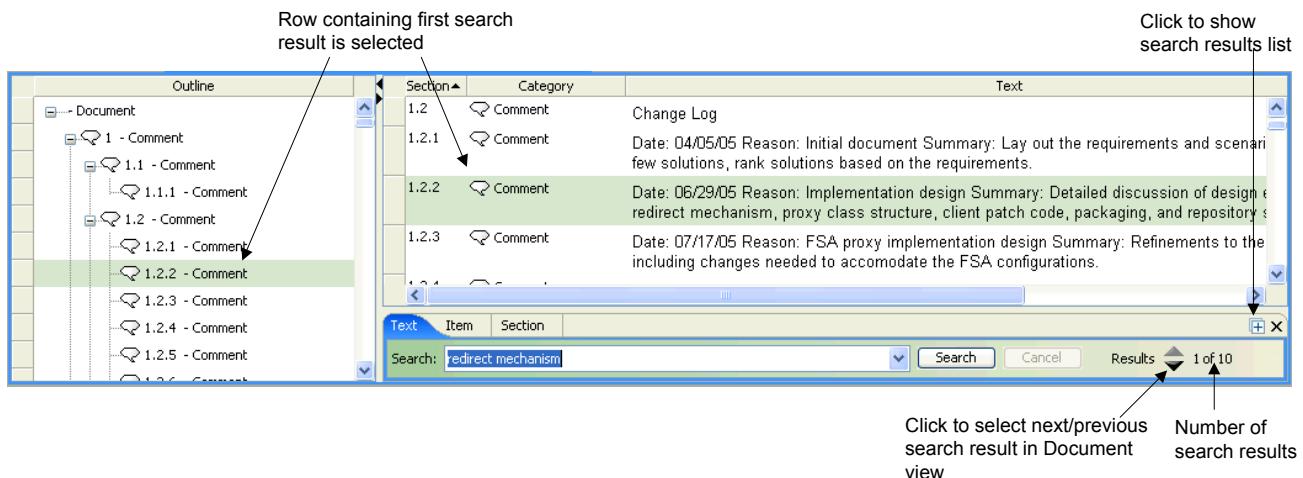
You can also enter descriptive text outside the brackets. The text you enter outside of the brackets will be displayed for each item in the Outline. Adding text can be useful for clarity; for example, a field label.

## Searching a Document

You can search a document in the Document view to find text, an item, or a section in the document.

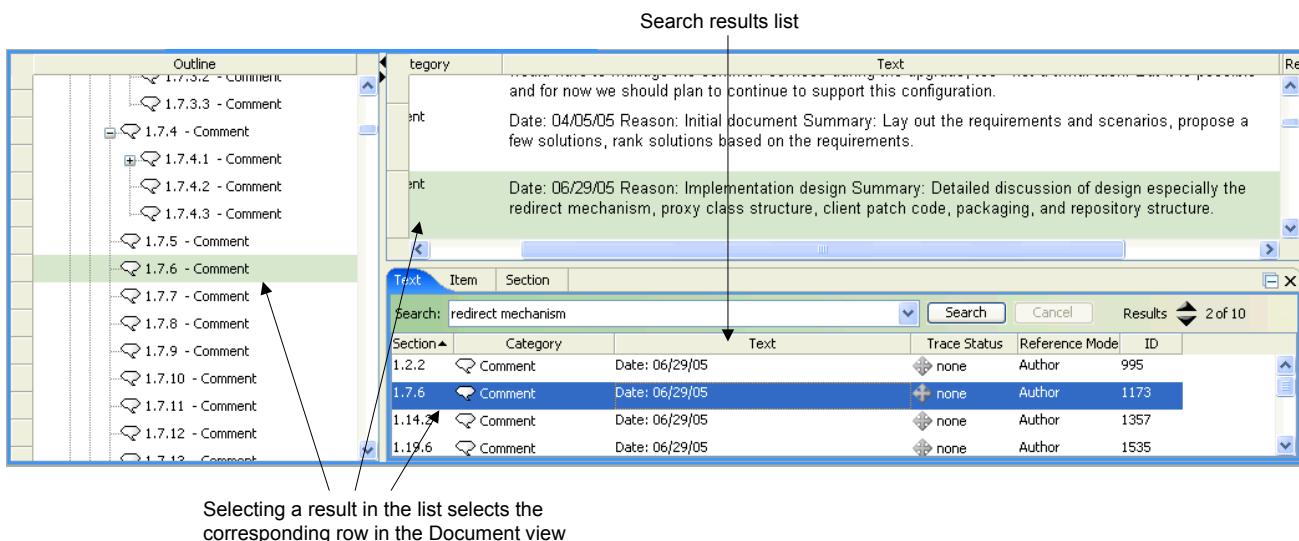
Menu	Shortcut Key	Type of Search
<b>View &gt; Search Text</b>	CTRL + F	Text
<b>View &gt; Search for Item</b>	CTRL + Shift E	Item ID
<b>View &gt; Search for Section</b>	CTRL + Shift S	Document section

The search panel displays at the bottom of the Document view Contents panel, with the tab for the type of search in focus. The search selects the first row in the document that matches the search criteria.



**NOTE** The document row is highlighted in the Contents panel and also in the Outline panel if the two panels are linked in the Document view options.

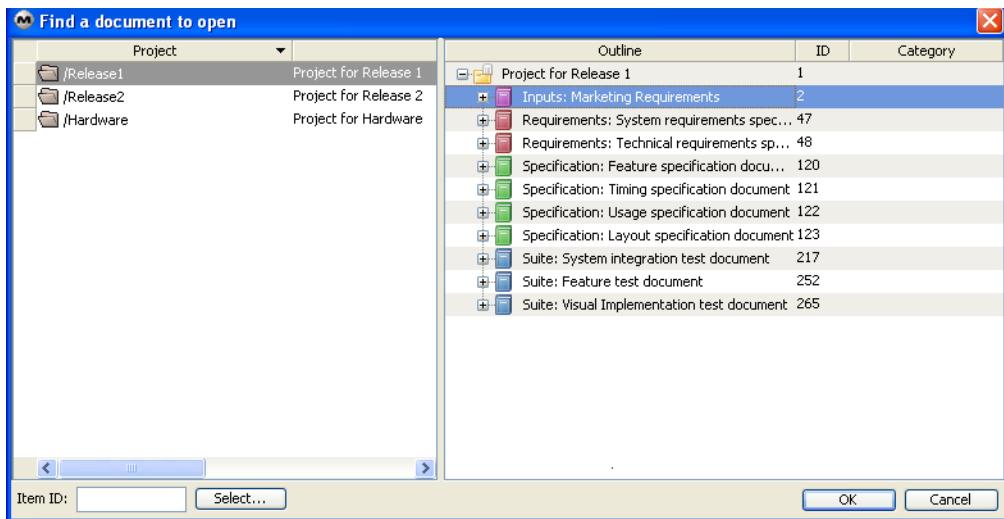
Searches for text or an item ID can return multiple search results. If there are multiple search results, use to move to the next or previous row that matches the search criteria. You can also click or drag the top of the search panel up to display a list of search results. You can click on a result in the list and the corresponding row is selected in the Document view.



# Document Finder

From the applicable MKS Integrity ViewSet, select **Document > Open**. The Finder dialog box displays documents by project based on a defined query. Projects display in the left panel and documents display on the right. When you select an item in the left panel, a tree of documents and associated content displays on the right. You can navigate through the documents in the right panel by using the nodes in the tree.

Click **Select** to change the list of documents based on a different query.



The Finder dialog box also displays when you insert or include documents. To learn how to add documents to an open document using the Finder dialog box, see “Adding Content to a Document” on page 194.

You can also enter an item in the **Item ID** field to display the root document associated with the document item. The selected item must be either a segment or a node. If it is a node, then the containing document opens.

# Working in the Document View

This section outlines the available symbols and indicators used to mark various actions in a document.

## Indicators and Icons

The following table lists some of the main indicators and icons that display when you perform actions on content. To learn how to perform that result in the following indicators and icons, see “Common Operations” on page 190.

Indicator	Description
	Signifies active, editable content.
Dotted highlight around content row	<ul style="list-style-type: none"><li>■ Red: Cut action.</li><li>■ Blue: Copy action.</li></ul> <p>These colors can be customized using the <b>View &gt; Options &gt; Colors</b> panel from the Document view.</p>
	Content is in the process of being saved. After content is saved, a new row displays after it. Rows are saved when a modification is made and you select a different context within the Document view. Blank rows are not saved.
New	New virtual row. Appears as first content item in new document. Displays after a row is inserted. Retains category of previous row. When you select a new row, a section number is assigned. Until you add content or select another section of the document, your changes remain unsaved.
Document messages	Pop up messages can be configured to indicate if mandatory fields are required or if you are entering a subdocument as you move through a document in the Content pane.

## Common Operations

The following tables outlines all of the available drag-and-drop, shortcut key, and other actions you can perform to manage content within the Document view.

### Rich Text

Rich Shortcut Keys	Action
CTRL+Z	(Multiple) Undo edit. Edits are committed to the database as soon as you save the selected item.
CTRL+Y	Redo edit.
CTRL+B	Bold toggle.
CTRL+U	Underline toggle.
CTRL+I	Italics toggle.
CTRL+K	Insert hyperlink.
CTRL+ click the link	Follow hyperlink.
TAB	Indent.
SHIFT+TAB	Decrease indent.
Double-click attachment	Edit attachment.
CTRL+⇒	Step to next word.
CTRL+⇐	Step to beginning of current word, or previous word.
CTRL+SHIFT+⇒	Highlight word.

## Rich Text

Rich Shortcut Keys	Action
CTRL+SHIFT+←	Highlight previous word.
Hold down SHIFT+ arrows	Highlight character by character in arrow direction.

## Document

Document Shortcut Keys	Action
CTRL+0	Open document using Finder dialog box.
CTRL+SHIFT+V	View document in Document view.
CTRL+SHIFT+R	Run report.
CTRL+SHIFT+B	Baseline document.
CTRL+N	Branch document manually.
CTRL+SHIFT+T	Change project document is associated with.
CTRL+SHIFT+ENTER	View (and potentially edit) properties associated with document (opens the Edit Properties dialog box).
CTRL+ click link	Follow MKS link in an editable cell in the Document view.
ALT+ click link	Follow MKS link in uneditable cell in the Document view.

## Content

Content Shortcut Keys	Action
CTRL+SHIFT+DELETE	Delete section.
INSERT	Insert content with the default category.
CTRL+SHIFT+X	Cut section.
CTRL+SHIFT+C	Copy section.
CTRL+ALT+V	Paste section into open document.
CTRL+ drag-and-drop	Copy content from one location to another in the content panel.
ALT+ drag-and-drop	Create trace relationship between content.
Drag-and-drop	Move content from one location to another in the content panel.
CTRL+S	Save.
ALT+ arrow	Navigate cells.
Arrow keys	<ul style="list-style-type: none"> <li>■ Left moves the field selection to the previous field in the same row.</li> <li>■ Right advances the field selection to the next field in the same row.</li> <li>■ Up moves the field selection to the same field in the previous row.</li> <li>■ Down moves the field selection to the same field in the next row.</li> </ul>
Save changes to row/item	<p>Inline field changes are saved automatically when you select a field in another row (MKS Integrity item) in the Document pane. To manually save changes to a field, or to save changes made in the Item Detail pane, Press CTRL + S or select <b>Item &gt; Save Item</b>.</p> <p>A pencil icon  displays in the view title bar if there are unsaved changes to an item in the view.</p>

# Creating a Document

The top-level document you create in this section will be the container for any content you create and modify. Top-level documents hold the project and document properties that will be applied for all of its nodes, including individual content items and subsegments.

**NOTE** Documents are typically created in a project or subproject that was set up by your administrator.

## Key Considerations

- Each section in a document is backed its own individual item. The document root is also represented by an individual document item.
- After a document is created the name displays in the Outline panel of the Document view and represents the top level of the document.
- Documents have specific project permissions, editability rules, and reference modes associated with them. You will not be allowed access to a document if you do not have membership in a group associated with the project the document was set up to belong to. If you cannot see a document, contact your system administrator to obtain the applicable permissions.
- You may not be allowed to move content in a document that requires a change order for modification. The document must be in an `Active` state or you must have an open change order assigned to you to permit moving content.

## To create a document from the Document menu in the GUI

- 1 Select **Document > Create > New** and select the applicable document type from the side menu. The list of available document types is determined by your administrator. The **Type Selection** dialog box displays.
- 2 Select the type of document you want to create.
- 3 Click **OK**. The **Project Selection** dialog box displays.

**NOTE** The types and projects that display in this dialog box are defined by your administrator.

- 4 Select a project to associate your document with. The **Create <Document Type>** window displays which resembles the Edit Item Details dialog box. Your administrator defines the fields you see in this dialog box. It is called *New* until you give it a name or specify Shared Text which will be used as the document title in the root of the Outline pane.
- 5 Define the document properties.

Your administrator may or may not have defined mandatory fields. As a result, at minimum, you need to add text to the **Summary** and **Shared Text** fields so that you can logically find your document by query. Both fields should contain the same data.

- **Summary:** A summary of the document as you want it to display in the dialog box title. For example, "Widget Requirements". Displays by default in queries.
- **Shared Text:** Shared text will display as the document identifier in the Outline panel. Shared Text also displays in subsegment references.

- 
- 6** Click **OK**. A new, empty document displays in the Document view. In the Content panel on the right, a blank row called *New* is available.

You can now add content to your document. For details about the ways you can create content in a document, see “Adding Content to a Document” on page 194.

To import a document from an external source select **Document > Import**. The Gateway Import Wizard displays.

To learn how to use the wizard, see the *MKS Gateway User Guide*.

## Creating a Document From a Template

You can create a new document from an existing document.

**To create a new document using an existing document as a template from the GUI:**

- 1** Select **Document > Create from template**. The Finder dialog box displays documents that are available for use as a template.
- 2** Select the document you want to use as a template for the new document from.
- 3** Click **OK**. The **Branch** dialog box displays, indicating that this document will be used as a template for the new one. All applicable fields and associated settings are copied over to the new document.  
At this stage, creating a document from a template is similar to branching. To learn about the options in this dialog box, “Branching Documents” on page 212.
- 4** Click **OK**. The **Item Details** for the new document displays allowing you to change the editable fields.

# Adding Content to a Document

You can add content to a document by:

- inserting an individual content item or row, or
- inserting a new or existing segment into a current document as a reference, or into another segment as a reference to the segment root

By default, segments are inserted as a reference within a document. That means they are not fully expanded to allow you to edit them in the context of the document. You must open the document to manage its contents.

Depending on how your administrator has configured your ViewSet, you can include subdocuments as a sequential part of the document, fully expanded for editability by including a new or existing segment, or into another segment fully expanded and editable from the segment with a reference to the segment root.

## ***Differences Between Included and Referenced Documents***

MKS Integrity allows you to create nested documents which are subdocuments inside of documents. This allows you to partition your content for logical, semantic, or performance reasons. Nested documents can be either inserted (as a reference) or included in other documents:

- **Insert** (reference): When a subdocument is inserted into a parent document, only the reference to the subdocument is exposed. You must open a subdocument in order to manage its contents.
- **Include**: When a subdocument is included into the parent, the entire contents of the subdocument are exposed as if they were a sequential part of the parent.

In a component-based environment, where an application or product is made up of smaller, more purpose-built components managed by separate teams, the inserted option is common. This allows the parent document to reference the content, but it does not require direct management of the content. Inserted documents are not exposed as part of the parent document.

After you insert or include a subdocument into a document, you can toggle the reference to either see the entire document expanded as a sequential part of the parent, or see the reference to the subdocument only in the content panel.

## Inserting a Row or Content Item

### To insert a row or content item

---

**CLI EQUIVALENT** im insertcontent

---

- 1 Open the document and find the applicable section where you want to insert content or a new row.
- 2 Select **Content > Insert**. A list of content types defined by your administrator displays in a side menu, for example, Heading, Comment, System Requirement, Test Group.
- 3 Select the appropriate option from the side menu list.

- 
- 4** Select the content type you want to insert. The new content displays above the row or section you selected. The **Category** field displays the type selected.

You can add text and format content using the inline editing and rich content options available from the **Format** menu and rich text field context.

---

**NOTE** You cannot insert content into a group document.

---

## Inserting a New or Existing Segment

### To insert a new or existing segment into a document from the GUI

---

**CLI EQUIVALENT** im insertsegment

---

- 1** Select the row location you want to insert the document and select **Content > Insert** to insert an existing segment.
- 2** Select **Document** if you want to insert an existing document. The Finder dialog box displays allowing you to select a document to insert into the current document content structure.

To insert a new document, select from the types of content available in the side menu. The **Item Details** view displays. Your administrator configures available content types. If no types were configured, select **New Document**.

After you select a new document, go to “Creating a Document” on page 192 to learn how to complete the Item Details view fields and configure the rest of the document.

To learn about the Finder, see “Document Finder” on page 189.

- 3** Select the document you want to insert from the Finder dialog box from the list of available documents by project.
- 4** Click **OK**. The Document is inserted. It is inserted as a reference, which means you have to open the document to modify its content. If you would like to view its entire contents in the Document view, select **Content > Toggle Include/Insert** to expand it in the current sequence.

---

**NOTE** Your administrator must configure which subsegments can be in documents. If you receive an error indicating that you cannot include subsegments into the document you are working with, contact your administrator.

---

Document hierarchies do not need to be structured alike. For example, you can insert/include a test document into a requirements document.

---

**NOTE** If you insert a segment into a group document, it is available as read-only.

---

### To insert an open segment into a target document as a reference from the GUI

---

**IMPORTANT** This is different from inserting a new or existing segment into a document in that you are inserting the *open* document into another document using the Finder dialog box for selection. The target document displays after you complete the Insert Into operation.

---

- 1** Open the document you want to insert into another document.

- 
- 2** Select **Document > Insert Into**. The Finder dialog box displays allowing you to select a document you want to insert the open document into.

To learn about the Finder, see “Document Finder” on page 189.

Document hierarchies do not need to be structured alike. For example, you can insert a test document into a requirements document.

- 3** Select the document you want to insert from the list of available documents by project.
- 4** Click **OK**. The open document now displays as a subsegment reference in the target document.

If you would like to view its entire contents in the Document view, select **Content > Toggle Include/Insert** to expand it in the current sequence.

## Inserting a Node from a Segment into an Open Segment

### To insert an existing node in another document into an open document from the GUI

- 1** Select the section in which you want to insert the node into and select **Content > Insert Content > From Document**. The **Finder** dialog box displays allowing you to select a node in a specific document to insert into the open document content structure.
- 2** Select the document you want to insert on the left side of the Finder dialog box from the list of available documents by project. The nodes display in the right panel.
- 3** Select the node you want to insert into your open document. Click **OK**.

The node is inserted into the selected section in your open document.

## Including a New or Existing Segment

### To include a new or existing segment into a document from the GUI

---

**CLI EQUIVALENT** im importsegment

---

- 1** Select the row location you want to include the segment in the document and select **Content > Include** to include an existing segment.
- 2** Select **Document** if you want to include an existing document. The Finder dialog box displays allowing you to select a document to include into the current document content structure.

To include a new document, select from the types of content available in the side menu. The **Item Details** view displays. Your administrator configures available content types. If no types were configured, select **New Document**.

After you select a new document, go to “Creating a Document” on page 192 to learn how to complete the Item Details view fields and configure the rest of the document.

To learn about the Finder, see “Document Finder” on page 189.

- 3** Select the document you want to insert from the Finder dialog box from the list of available documents by project.

- 
- 4 Click **OK**. The Document is included as an expanded, sequential part of the parent document. You can edit the segment as if it were a part of the document.

If you would like to see the document in the tree as a reference only, select **Content > Toggle Include/Insert**.

---

**NOTE** Your administrator must configure which subsegments can be in documents. If you receive an error indicating that you cannot include subsegments into the document you are working with, contact your administrator.

---

Document hierarchies do not need to be structured alike. For example, you can include a test document into a requirements document.

---

**NOTE** If you include a segment into a group document, it is available as read-only.

---

### To include an open segment into a target document from the GUI

---

**NOTE** This is different from including a new or existing segment into a document in that you are including the *open* document into another document using the Finder dialog box for selection. The target document displays after you complete the Insert Into operation.

---

- 1 Open the document you want to include into a document you will choose from the Finder dialog box.
- 2 Select **Document > Insert Into**. The Finder dialog box displays allowing you to select a document you want to include the open document into.

To learn about the Finder, see “Document Finder” on page 189.

Document hierarchies do not need to be structured alike. For example, you can insert a test document into a requirements document.

- 3 Select the document you want to include the document into from the list of available documents by project.
- 4 Click **OK**. The open document now displays as a subsegment reference in the target document.

If you would like to view its entire contents in the Document view, select **Content > Toggle Include/Insert** to expand it in the current sequence.

## Viewing and Editing Document and Content Items

You can display detailed item information for the item selected in the Outline or Contents pane. You can display either the Item Detail view or the Edit Item Detail view as a docked part of the Document view.

To display item detail information as a docked part of the Document, select **View > Options**, and then select **Show item detail view** on the **General** tab of the **Options** dialog box.

To display editable item detail information as a docked part of the Document view, select **View > Options**, and then select **Show item edit view** on the **General** tab of the **Options** dialog box.

Editing items through the **Edit Item Details** view instead of the Content pane allows you to edit all editable fields and correct errors in fields that do not display in the Content pane. If the **Edit Item Details** view is not displayed, it will display automatically whenever you need to: resolve mandatory fields in the Item Details view, problems with field relationships, event triggers, and editing conflicts.

---

Changes to content are reflected in the Item Details view. Changes made in the Edit Item Details view are reflected in the Content pane if the change made to that field is displayed as a column. When the item is modified in either pane the view title bar shows the changed symbol (\*) and a  displays beside the list item and at the top of the detail pane until the changes are saved. Changes are saved when you move to another item in the document.

---

**NOTE** In the Document view, you can select and edit one row at a time. If you select multiple rows in a Document view, you cannot edit any of the rows.

---

---

# Modifying Content

After content is added to a document, you can modify it in the following ways:

- copying content within and between documents (see “Copying Content” on page 199)
- move content around a document or delete a section (see “Restructuring Content in a Document” on page 201)
- change a share or reuse reference mode (see “Changing the Reference Mode” on page 201)
- toggle a subsegment reference (see “Toggling a Subsegment Reference” on page 201)

## Copying Content

You can copy content within the same document or between documents using standard GUI copy and paste operations.

If your administrator has configured your ViewSet appropriately, you can copy content between different documents of the same type using the **Paste Special** gesture.

### To copy content in the GUI

- 1 Open the applicable document.
- 2 Select a row or section in the Content panel and select **Content > Copy**. The selected row is highlighted.
- 3 Select an insertion point and select **Content > Paste** to paste the row into the target section. The insertion point could be in a different document, but the target document must be the same type as the source document.

The default reference mode for the type is applied.

### To copy multiple content items in the GUI

- 1 Open the applicable document(s).
- 2 Press CTRL and select the multiple rows in the Content panel or the Outline panel and press CTRL+C, or select **Content > Copy**. The selected rows are highlighted. Selecting a node to copy from the Document view copies just the selected items. Selecting a node to copy from the Outline panel copies all items in a node’s structural relationship list.
- 3 Select the insertion point in the existing or target document and select **Content > Paste**. The insertion point could be in a different document, but the target document must be the same type as the source document. If the copied nodes belong to the same segment in the target segment they are branched and the content and the shared items are also copied.

---

**NOTE** Selecting a node to copy from the Document view results in just the selected items being copied. Selecting a node to copy from the Outline panel results in all items in the node’s structural relationship list being copied.

---

If the copied nodes span multiple segments, the **Paste Special** dialog box displays, allowing you to choose whether or not you want to paste the content items for the purposes of share, reuse, or just a regular GUI copy. The **Include Traces** checkbox allows you to specify to carry over all traces associated with the children of the copied nodes.

---

## To copy content within and between documents for the purposes of reuse

When you copy content within the same document, you are performing a standard GUI copy operation. Copying between documents is generally for the purposes of reuse.

- If you want to do a standard copy either within a document or to another document, press **CTRL+V**.
- If you want to copy for the purposes of reuse, click **Content > Paste Special**.
  - 1 Open the applicable document.
  - 2 Select a row or section in the Content panel of the document you want to copy from and select **Content > Copy**. The selected row or section is highlighted. Selecting a node to copy from the Document view results in just the selected items being copied. Selecting a node to copy from the Outline panel results in all items in the node's structural relationship list being copied.
  - 3 Select an insertion point in the target segment and select **Content > Paste Special** to paste the row.  
If the copied nodes span multiple segments, the **Paste Special** dialog box displays, allowing you to choose whether or not you want to paste the content items for the purposes of share, reuse, or just a regular GUI copy. The **Include Traces** checkbox allows you to specify to carry over all traces associated with the children of the copied nodes.

---

**NOTE** If you do not select a new reference mode, you can change the reference mode at another time using **Content > Toggle Share/Reuse**. However, you can only change it between Share and Reuse; Author is not an option.

---

## To copy historical content in the GUI

---

**NOTE** You cannot copy content into a historical view of a document.

---

- 1 Open the historical version of the document or content using a query or by selecting **Document/Content > Historical > View As Of**.
- 2 Select an option from the As Of list.
- 3 Click **OK**. The historical version of the document displays based on the criteria you selected.
- 4 Open the document you want to paste the historical document into.
  - To copy an entire section and its children, select the section from the Outline panel and select **Content > Copy**.
  - To copy the content only, select the content you want to copy from the Document panel and select **Content > Copy**.

Selecting a node to copy from the Document view results in just the selected items being copied. Selecting a node to copy from the Outline panel results in all items in the node's structural relationship list being copied.

- 5 Choose a selection point in the historical document or in another document and select **Content > Paste**.

Subsegment references copied from historical documents are pasted into the target segment as a shared item instead of a segment. The shared item text in the top level node in the Outline panel displays as an As Of hyperlink to the original segment. You can select this link to go to the original segment and modify it. The Author reference mode is applied to the newly created content.

If you are copying and pasting in the standard GUI way, the content displays at the applicable insertion point.

# Restructuring Content in a Document

You can restructure the content in a document by moving or deleting content.

## To move content in the Document view

**CLI EQUIVALENT** im movecontent

- To move content up, select **Content > Move > Up**.
- To move content down, select **Content > Move > Down**.
- To move content farther in, select **Content > Move > Increase Section Level**
- To move content farther out, select **Content > Move > Decrease Section Level**

## To delete a row or section

**CLI EQUIVALENT** im removecontent

Select the row you want to delete in the Content panel and select **Content > Delete** or press the **Delete** key. When content is deleted from a document, all trace relationships going to the content are also removed.

**NOTE** You can delete multiple sections but they must all be at the same section level within the same document. For example, sections 4.1, 4.2 and 4.5 can be selected together, while sections 4, and 4.1 cannot.

You cannot delete content that has a share reference as other documents are dependent.

The following option may not be available to you depending on how your administrator configured the **Content** menu.

Select the section node you want to delete in the Outline panel of the Document view and select **Content > Delete Section**. The operation deletes the selection and its corresponding child content (subsections and other content) from the document.

# Changing the Reference Mode

**CLI EQUIVALENT** im refmode

You can change the reference mode on content by selecting the content you want to change the sharing for by selecting **Content > Toggle Share/Reuse**, and choosing between **Reuse** or **Share**.

**NOTE**

- You cannot change the reference mode for meaningful content in a group document or for content with a reference mode of Author. For more information on document model terms and concepts, see “What is a Document?” on page 181.
- By default you will reuse from the node from which you copied.

# Toggling a Subsegment Reference

You can expand or hide subsegment content in a segment root by selecting the applicable subsegment from the Document List view and choosing **Content > Toggle Include/Insert**.

---

The associated subsegment content is no longer visible under the subsegment. Select the subsegment again and choose **Content > Toggle Include/Insert** again to see content expanded.

---

# Finding and Viewing Documents

---

**CLI EQUIVALENT** im viewsegment

---

The following table outlines the ways you can search for and view documents and content.

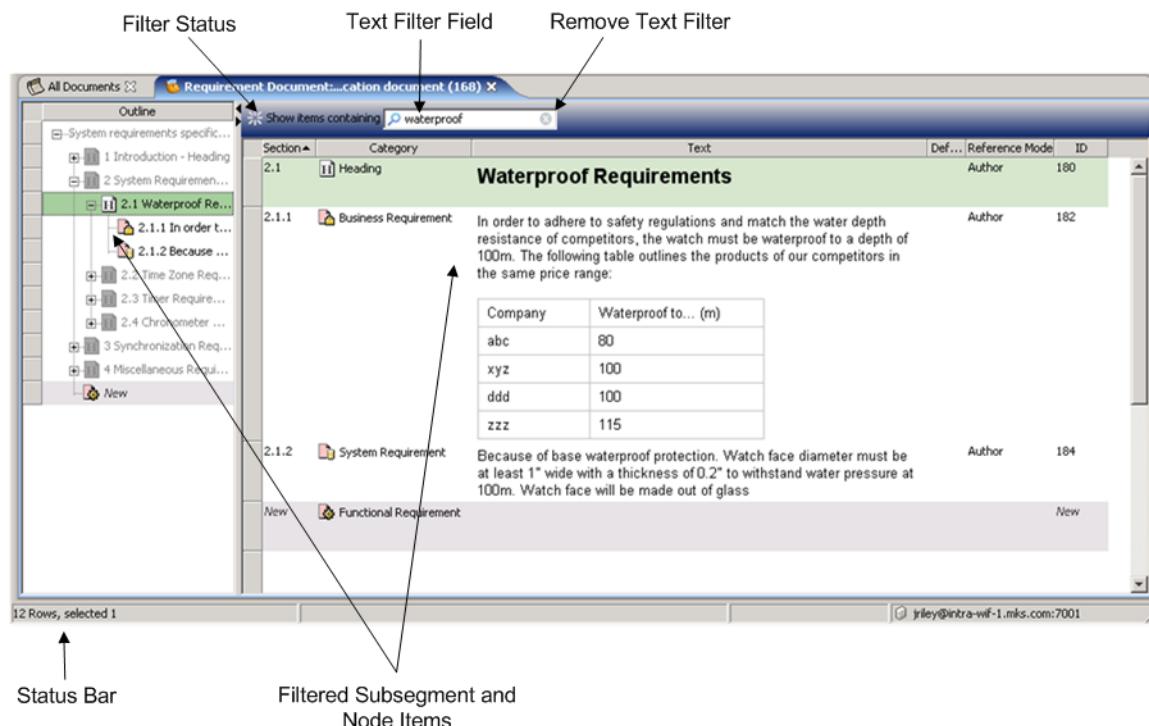
To...	Do this...
Find and open a document	Select <b>Document &gt; Open</b> . The Finder dialog box displays, allowing you to select an existing document. You can also search on a document ID. Select <b>Document &gt; Recent</b> . A side menu of the most recent documents you have viewed displays. Select the one you want to view and it will display in the Document view.
View documents by query	Select <b>Item &gt; View Items</b> to display items as documents in an Items view based on the defined query. You can also view documents and content by query from the Finder dialog box <b>Note:</b> To prevent performance issues, MKS recommends you do not filter on Document ID in a query.
Viewing a document <b>As Of</b> a specific date and or label, branch, or user edit	Select the document item you want to view in the Items view and select <b>Item &gt; Historical &gt; View Item As Of</b> . The <b>View Item As Of</b> dialog box displays allowing you to view the document as of a specific date, label, branch, or user (modification). The item representing the document displays in the Item Details view. Historical items are indicated by the As Of date in the item header.

# Filtering in the Document View

In addition to filtering provided by the Finder dialog box for documents, the Document view contains a filter bar that allows you to filter by visible text in the document. The filter criteria creates a sentence in the filter bar that indicates what you are looking for, for example: Show items containing waterproof.

## Filter Bar: User Interface Components

The following is an example of an active filter bar in the Document view:



The filter bar includes the following user interface components that allow you to search in the Document view:

Component	Description
Filter Status	After you type the text that you want to filter by, the filter status icon animates to indicate progress. When the text filter is applied to the Document view, the color of the filter bar changes color. When the text filter is removed, the filter bar reverts to the default color.
Text Filter Field	<p>Filters items by visible text in the Document view.</p> <p>Type the text that you want to filter by. The list of items is filtered when there is a pause in typing or when you finish typing.</p> <p>Note the following:</p> <ul style="list-style-type: none"><li>■ The text filter field is case-insensitive.</li><li>■ To filter for an exact text string, use double quotes, for example, "water depth". If you do not specify double quotes, MKS Integrity searches for items containing all of the words in the text string in any order.</li><li>■ You can also filter by numeric values. A text filter that includes a section ID returns results; however, it does not return the specific section ID you are searching for. For example, if you type 2.1, sections 2.1, 2.1.1, and 2.1.2 display.</li></ul>
Remove Text Filter	Clicking  clears the text filter field, removing the applied text filter.

Component	Description
Status Bar	Displays the number of items matching the text filter and the selected items.
Filtered Subsegment and Node Items	Text filtering removes non-matching rows from the content pane only. All non-matching rows in the outline view remain visible to retain a complete view of the outline; however, the non-matching rows are greyed out.

### ***Key Considerations***

- The shortcut key for filtering in the **Document** view is **CTRL+SHIFT+F**.
- Only documents with less than 3,000 rows can be filtered. This value is non-configurable.
- If **Link with detail** or **Link with outline** are disabled in the **Options** dialog box, a selection in the content pane is not preserved after applying a text filter.
- The delete, cut, copy, and paste operations are available in a filtered view; however, you can only select rows matching the filter as targets and non-matching rows in the outline pane.
- If you select a row in a filtered view for a cut/copy operation and then change the filter, the paste operation is still available (even when the cut row is filtered out).
- By default, if you paste content in the content pane that does not match the current text filter, a dialog warns you that the pasted content will not display. You can disable this warning from the **Options** dialog box.
- In a move operation with a text filter applied, the rows above and below the selected target of a move operation may be filtered out in the content pane, making the results unclear. In this scenario, MKS recommends switching to the outline pane to perform move operations.
- You can use inline editing to edit matching rows in the view. If you change the value in a row that matched the filter, the row remains visible until you re-apply the filter.

---

# Printing Documents

From the Items view, with a document selected, select **View > Print**. You are presented with options for printing the selected document.

The document prints in the form of the defined report if your administrator defined one on the segment type. If a report was not defined, the contents of the Content panel on the right print.

---

**NOTE** The Web interface does not support the **Document** view.

---

---

# Viewing and Editing Document or Content Properties

To view and edit document properties, select the applicable document or content item from the Document List view, and select the **Document** menu or **Content** menu > **Properties**. The **Item Details** view displays document property details.

The types of document properties you can define or view depend on how document types are defined by your administrator. **The following is a sample selection of typical fields available in the Item Details view showing typical document properties:**

- **Summary** of the document
- **Shared Text** (often the same data entered in the **Summary** field) which defines the name of the segment root that displays in the first row of the Outline panel
- nodes it **Contains**
- nodes it is **Referenced By**
- **Project**
- **State**
- **Shared Attachments**

To specify values for certain fields, you must be familiar with using the data filter.

---

# Viewing Document Model Relationships

---

**CLI EQUIVALENT** im relationships

---

To display relationships in a Document:

- In the GUI, select one or more items in the **Items** view or **Item Details** view, then select **Item > View Relationships**.
- From within an open document, then select **Item > Relationships**.

---

# Changing the Document Project

---

**CLI EQUIVALENT** im changesegmentprojec

---

You can change the project a document belongs to.

- 1 From the document list view, select the document you want to change, and then select **Document > Change Project**. The **Project Selection** dialog box displays, allowing you to select a new project.
- 2 Select the new project. The new project is applied to the segment root and its associated nodes.

Your administrator typically sets up projects you can link the document to. For more information about setting up projects, see the *MKS Integrity Server 2009 Administration Guide*.

# Baselining Documents

---

## CLI EQUIVALENT im baseline

---

A baseline is a meaningful point in a document's history. You would baseline an item or group of items to mark a release date or milestone.

Baselines are denoted using labels. Labels provide a name for baselines and associate them with a specific point in time. Your administrator determines the types of items that can have labels and who can add labels to items.

This section describes labelling in the context of using a label to baseline a document.

To learn how to enable or disable type properties, see the *MKS Integrity Server 2009 Administration Guide*.

---

**NOTE** Labels may be any alphanumeric string without control characters. They may also include spaces and punctuation.

---

## Key Considerations

- A document baseline captures the current state of all content.
- Your administrator determines who can baseline documents.
- You can baseline one document or multiple documents at once.

## To baseline an individual document from the Document view

- 1 Open the document you want to baseline and select **Document > Baseline**. The **Baseline** dialog box displays containing the Document ID of the document you are branching in the header.
- 2 Type the name of the **Label** for the baseline, for example, Project ABC System.
- 3 Select the **As Of** option from the list to associate the branch with a specific point in time.

The following table lists the options for branching documents **As Of**:

To branch a document As Of...	Do this...
Now	Select <b>Now</b> to baseline the current version of the document.
Branch	<b>Note:</b> This option is only available if you are baselining a document that was branched or is a branch document. Select the applicable branch from the list. You can type some text in the <b>Show Branches Containing</b> text box to display edits containing specific text.
Label	Baseline as of another baseline. Two names for the same version of the doc.
Edit	Select the applicable edit from the list. You can type some text in the <b>Show Edits Containing</b> text box to display edits containing specific text.
Date	Select <b>Date</b> . A calendar displays allowing you to select a date to baseline the document as of. The document must have existed at the date and time you select or else you will get an error.

## To baseline multiple documents from the Document view

- 1 Find the documents you want to baseline and select **Document > Baseline**. The **Baseline** dialog box displays.

- 
- 2** Type the name of the **Label** for the baseline, for example, Project ABC System.
  - 3** Select the applicable **As Of** option from the list. The default date is the current date and time (Now). If you are baselining a historical item, the associated date and time displays by default in the **As Of** field.

The following table lists the options for baselining documents **As Of**:

To baseline a document As Of...	Do this...
Now	Select <b>Now</b> to baseline the current version of the documents.
Date	Select <b>Date</b> . A calendar displays allowing you to select a date to baseline the documents as of. The document must have existed at the date and time you select or else you will get an error.

The new baseline label is available in the list on the **Labels** tab in the Item Detail view or by viewing document properties.

---

# Branching Documents

---

**CLI EQUIVALENT** im branchsegment

---

Branching allows you to support concurrent development of projects having very similar documents. Branching one or more documents creates a new copy of each existing document so that you can use them to begin work on a new project or document.

## **Key Considerations**

- In order for you to branch a document you must have permission to create documents and edit document properties in the target project.
- You can branch one document or multiple documents at once.
- Your administrator determines who can branch documents and how and when traces are propagated into the newly branched document(s).
- You can view the details of an item branch on the **Branches** tab of the Item Detail view.
- Once a branch is created, it cannot be undone.
- The parent item remains unchanged during a branch operation. The branched item copies current field information from the parent item. A new entry is added to the parent item **Branches** tab.
- Your administrator configures which fields in the type will be copied to the new item. Fields are copied regardless of whether you have permission to view or edit them.

## **To branch an individual document from the Document menu**

- 1 Select the root of the document(s) you want to branch and select **Document > Branch**. The **Branching** dialog box displays containing the ID of the document you are branching in the header.
- 2 **Select a Destination Project** and click **OK**.

- 
- 3 Select the **As Of** option from the list to associate the branch with a specific point in time.

The following table lists the options for branching documents **As Of**:

To branch a document As Of...	Do this...
Now	Select <b>Now</b> to branch the current version of the document.
Branch	<p><b>Note:</b> This option is only available if you are branching a document that was branched or is a branch document.</p> <p>Select the applicable branch from the list. You can type some text in the <b>Show Branches Containing</b> text box to display branches containing specific text.</p>
Label	<p>Select <b>Label</b>. A list of labels associated with the item displays.</p> <p>Select the label you want to branch the item as of.</p> <p><b>Note:</b> This option is unavailable if a document has not been labelled or baselined at least once.</p>
Edit	<p>Select <b>Edit</b>. A list of user names associated with modifications to the document displays.</p> <p>Select the username and details you want to create the branch as of.</p>
Date	Select <b>Date</b> . A calendar displays allowing you to select a date to branch the document as of. The document must have existed at the date and time you select or else you will get an error.

---

**NOTE** When branching a document based on a point in the past, subsegments are replaced by a shared item that has an As Of hyperlink to the segment root.

- 4 Click **OK**. The **Item Details** for the branched document displays.

---

**NOTE** When you branch a document it creates a copy of an entire document without copying the actual text content, linked images, or attachments. The text content is shared from the original document.

#### To branch multiple documents from the Document menu in the Items view

- 1 Select the documents you want to branch and select **Document > Branch**. The **Branching** dialog box displays the number of documents you are branching in the header.
- 2 In the **Select a Destination Project** box, select the project you want the branched documents to be a part of.

---

**NOTE** You can only apply one project to all branched documents.

- 3 Select the **As Of** option from the list to associate the branch with a specific point in time.

The following table lists the options for branching documents **As Of**:

To branch a document As Of...	Do this...
Now	Select <b>Now</b> to branch the current version of the documents.
Date	Select <b>Date</b> . A calendar displays allowing you to select a date to branch the documents as of. The documents must have existed at the date and time you select or else you will get an error.

- 
- 4 Click **OK**. The branched documents display in a list in the **Item Details** view.

You can edit the documents you branch using the inline edit option.

---

**IMPORTANT** When you branch documents, traces may or may not be propagated automatically. How traces are propagated to branched documents depends on how your administrator configured the behavior of document branching in your environment. For example, an individual document with traces between items contained in the document may be set up to propagate automatically. However, if the document you are branching is more complex and has traces to items within other documents of varying types, for example, test cases, traces may have to be propagated manually after the branch occurs. To learn more about how and when to propagate traces manually, see “Propagating Traces to Branched Documents” on page 218.

---

## Viewing Branch Details

The **Branches** tab in the Edit Properties or Item Detail view allows you to see a full history of branched documents. This tab displays the branch ID which allows you to view a history of documents that have been branched from the active document, a summary of the branch if one was created, the user who copied or edited the document, and the date it was branched.

The **History** tab on the Item Detail view displays a log of all changes to an item.

# Managing Trace Relationships

A trace relationship is any relationship between two content items; for example, a trace between documents and content of different types. Trace relationships are configured by your administrator. To create a trace relationship, you only need to specify the start and end points; the specific relationship is created automatically if it has been configured by your administrator.

To learn more about trace relationships in the context of other document relationships, see “Relationships in the Document Model” on page 183.

For example, John is working on document ABC and creates a change order requesting work. Bob, a manager, assigns the change order to document XYZ. Dan, another analyst, modifies document XYZ to satisfy the change order, then creates traces back to document ABC to illustrate how the changes reflect the requirements stated in the originating document.

Traces are used to show the backward and forward relationships between other artifacts in the system. There can be multiple traces to or from the same content. Like other content, back trace information can usually be updated only when the content is editable.

---

**NOTE** To establish a trace, a relationship that reflects the type of trace being created must be created between content.

---

## To create a trace from the Content menu in the GUI

- 1 Select a requirement and select **Relationships > Create Trace**. The **Find item(s) to Trace To** dialog box displays.
- 2 Navigate through content and documents to select the item to trace to.
- 3 Click **OK**. The proper trace direction and trace field is automatically determined.

---

**NOTE** The types of content that you can create a trace to are defined by your administrator. Not all start and end points can have trace relationships. You can create pre-configured relationships between any two types in the system, not just between content types. Your administrator may need to configure the application to take advantage of this feature.

---

## To create a trace using drag-and-drop

- 1 Open the documents that contain the items you want to create traces between. Use the view options to place the documents where you want them (top and bottom or side by side).
- 2 Hold ALT and drag-and-drop one of the items to the item you want to create the trace relationship with until you see **As Relationship** display with a black arrow. The **Move Relationships** dialog box displays.

---

**NOTE** The direction of the drag-and-drop does not matter because the configuration automatically determines the proper relationship field and direction to trace.

---

- 3 Select the **Relationship Field** you want to use to create the trace relationship.
- 4 Click **OK**.

---

## Viewing Trace Relationships

Depending on how your administrator has configured your ViewSet, you can view trace relationships between content items in a variety of ways; for example, upstream, downstream, or peer traces. You can also view other relationships such as authorizations, spawned tasks, and references. For complete details about trace view options, see the documentation included with the ALM solution download, available on the MKS Customer Community (<http://www.mks.com/community>).

---

## Running Current and Historical Reports on Documents

You can run current or historical reports on documents.

To run a historical report, select the document you want to run the report on from the Document view and select **Document > Run Report As Of**. The **Run Report As Of** dialog box displays allowing you to select to run a report on a historical document by date, branch, or (user) edit. You can also choose to run the report as of the current date.

# Propagating Traces to Branched Documents

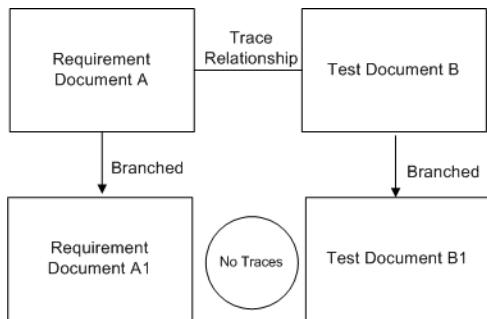
The fields that are copied when you branch a document are determined by your administrator. MKS recommends that trace relationship fields are not automatically copied during branching. This enables you to control which trace relationships get copied to the branched document using the propagate traces command.

For more information on trace relationships, see “Managing Trace Relationships” on page 215.

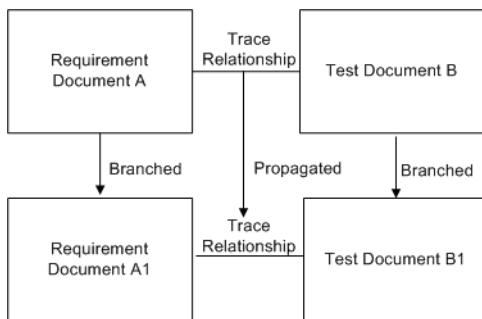
For more information on branching a document, see “Branching Documents” on page 212.

## ***Example 1: Propagating Traces Between Two Branched Documents***

Planning is started for Version 2 of a particular software component. Version 1 of this component has a requirements document and a trace relationship to a related test node. When the analyst starts work on the requirements for Version 2, he branches the original requirements document. Because the administrator has not set up trace relationships to be copied during a branch operation, the trace relationship to the test document is not copied. When the test author starts work on the test plan for Version 2, she branches the original test document. The trace relationship to the requirements document is not copied.



Before the requirements analyst publishes the requirements document for Version 2, he wants to have a related test plan. He needs to copy the trace relationship that exists between the requirements and the test document in Version 1 to Version 2. He does this using the propagate traces command.

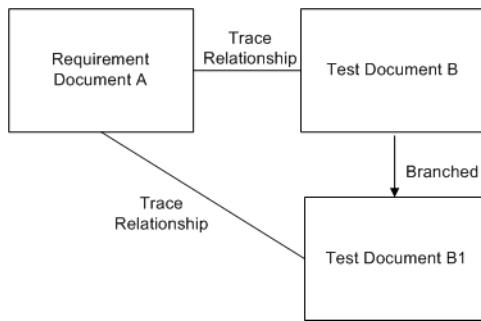


This example illustrates a simple propagation of a trace relationship between two branches. For more complex examples, see the ALM section of the MKS Customer Community Knowledge Base found at <http://www.mks.com/community>.

---

### **Example 2: Propagating an Existing Trace Relationship to a Branch**

You can copy an existing trace relationship between two documents to a branch of one of the documents. This creates a trace relationship from one version of the document to two versions of the related document. For example, a trace relationship exists between a requirement document and a test document, a new test case for the requirement has been created by branching the existing test case, and you want to create a trace relationship from the requirement to the new test case.



#### **How Traces Are Propagated**

When propagating traces, you start by selecting the document that you want to propagate traces to. In the Propagate Traces wizard, this is called the **Copy To Start** document. In Example 1, you would be selecting Requirement Document A1 as the document you want to propagate traces to. You then specify the following:

- 1 The document you want to propagate traces from, for example, the original requirements document that was branched from. In the Propagate Traces wizard, this is called the **Copy From Start** document. In Example 1, this would be Requirement Document A.
- 2 The document that has a trace relationship with the document you want to propagate traces from. In the Propagate Traces wizard, this is called the **Copy From End** document. In Example 1, this would be Test Document B.
- 3 The document you want to propagate the trace document to, for example, a test document in the branched requirements document. In the Propagate Traces wizard, this is called the **Copy To End** document. In Example 1, this would be Test Document B1.
- 4 The trace relationship you want to propagate to the target document, for example, the **Tests** relationship.

This process creates two maps:

- One relating each entry in the Copy to Start document to a corresponding entry in the Copy from Start document.
- One relating each entry in the Copy to End document to a corresponding entry in the Copy from End document.

MKS Integrity then propagates the trace relationships. For each Copy From Start document that traces are being propagated from, MKS Integrity determines all the trace relationships that go to the Copy To Start document. For each trace relationship, MKS Integrity finds the corresponding Copy From End and Copy To End documents. If the trace relationship does not exist in the Copy to End document, MKS Integrity creates it.

---

**IMPORTANT** Traces are propagated correctly even if the document contents have been edited or moved.

---

## Suspect Traces

After propagating trace relationships using this command, any of the newly created trace relationships that are considered to be suspect are flagged for your review. Traces are flagged as suspect in the following situations:

- If a suspect flag exists on the trace relationship being copied (even if the trace relationship already exists on the Copy To End document and is not suspect)
- If the Copy From Start document's shared item ID is not the same as the Copy To Start document's shared item ID, or if the Copy To End document's shared item ID is not the same as the Copy From End document's shared item ID, then the new trace is marked suspect on the Copy To End document.

## Key Considerations

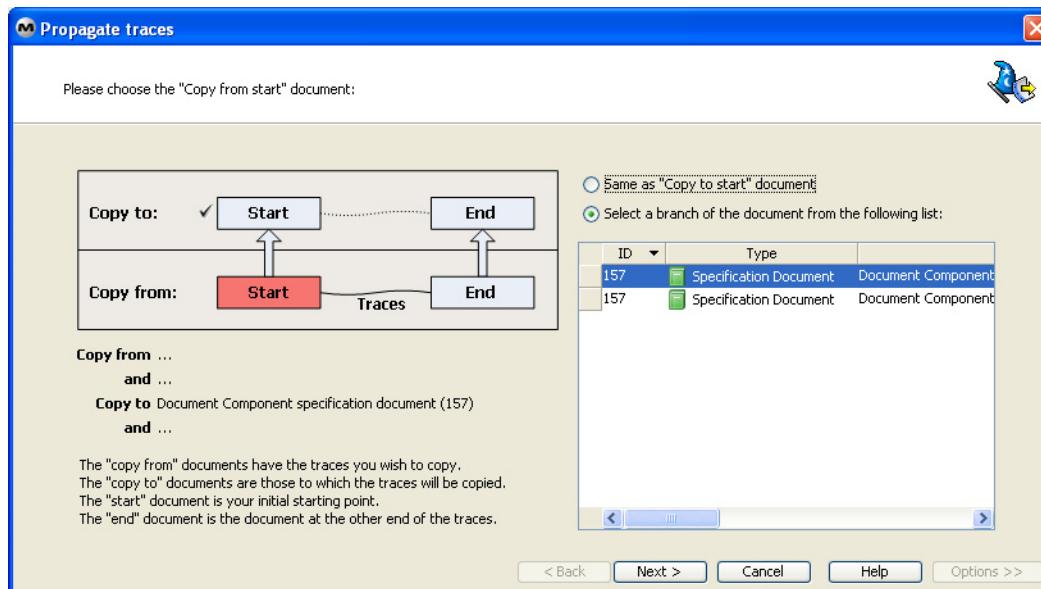
- You can only propagate traces between one pair of documents at a time.
- Traces are propagated as of the current time, not as of time the branch was created. Therefore, any changes made to the traces since the branch was created are propagated.
- When new trace relationships are created through trace propagation, it is noted in the document node history and the document history.
- When propagating traces in the GUI, the Copy To Start document and the Copy From Start document must be related to each other in the branching history. The CLI command `im propagatetraces` does not have this restriction.

## To propagate trace relationships to branched documents

- 1 Open or select the document that you want to propagate trace relationships to.

**NOTE** You must not open or select a historical version of the document. It must be as of now.

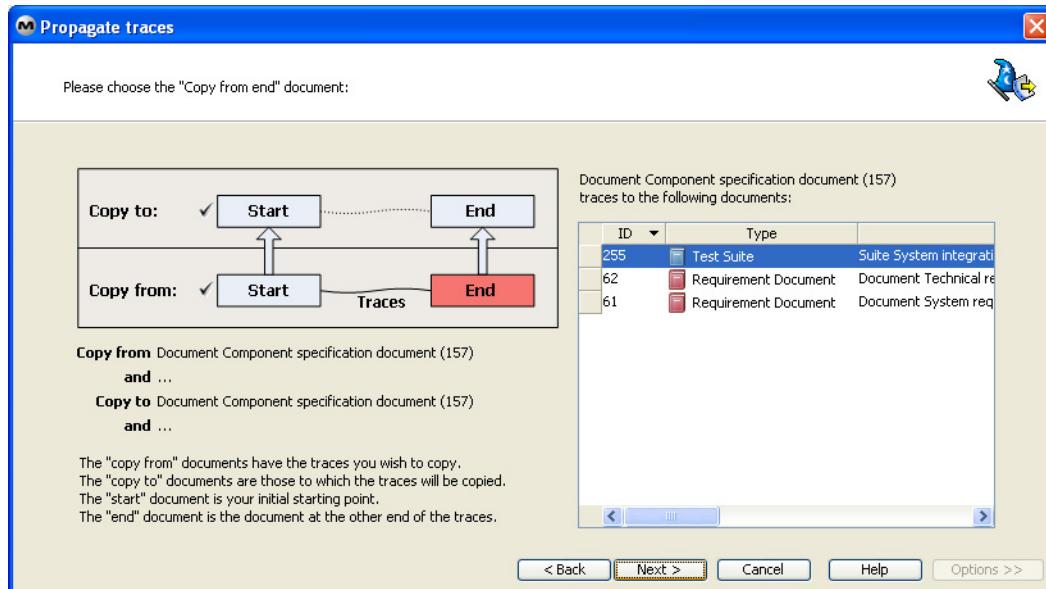
- 2 Select **Document > Propagate Traces To**. The first panel of the **Propagate Traces** wizard displays.



- 3 Select the document that you want to copy traces from:

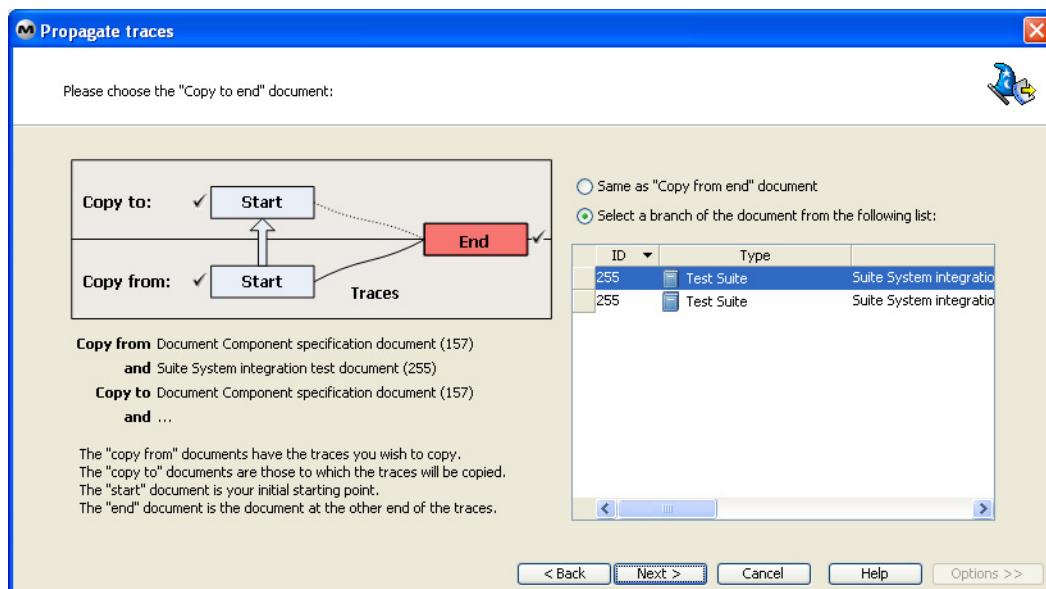
- 4 Select the document from which you want to copy the traces. The document you select should have trace relationships that you want to propagate to the document you selected in Step 1.
- To copy trace relationships from the document you selected in Step 1, select **Same as "Copy to Start" document**.
  - To copy trace relationships from a branch of the document you selected in Step 1, choose it from the list. The most recent predecessor is selected by default.

Click **Next**. The second panel of the **Propagate Traces** wizard displays.



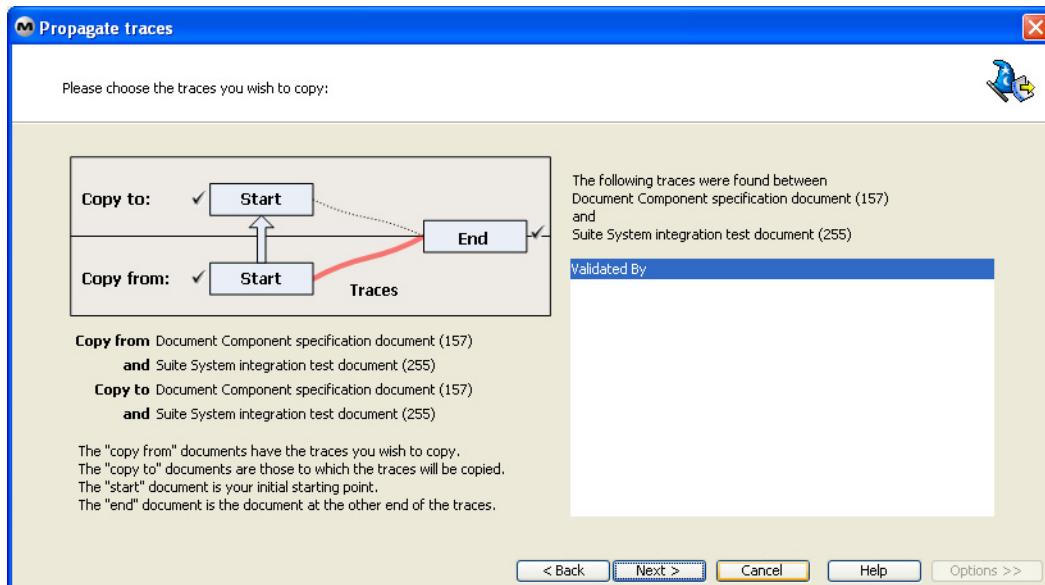
- 5 Select the document that has a trace relationship with the document that you want to copy traces from. You can select a document that has any type of trace relationship.

Click **Next**. The third panel of the **Propagate Traces** wizard displays.



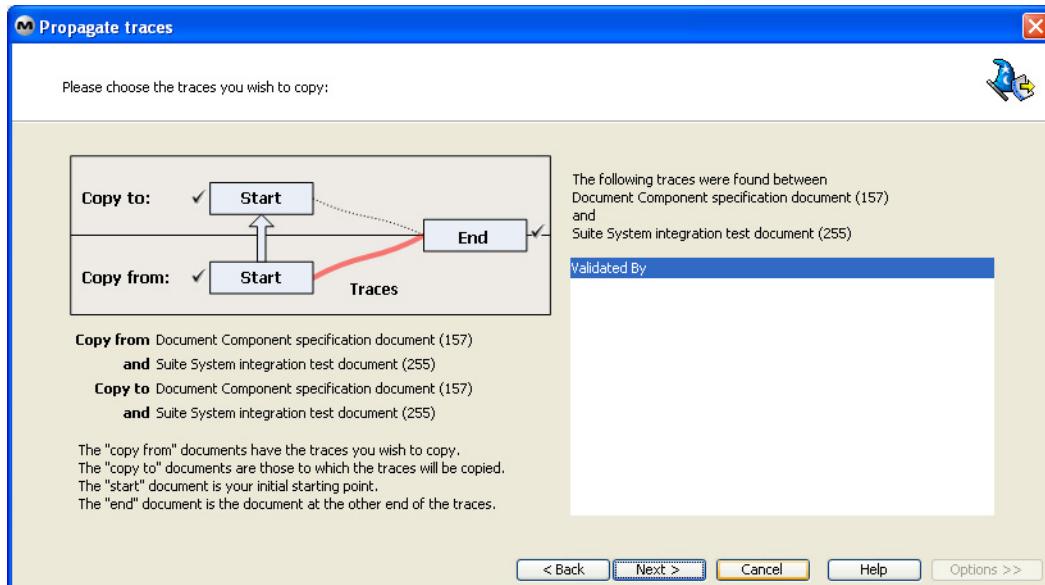
- 6** Select the document that you want to create a trace relationship for:
- Create a trace relationship for the document you selected in Step 4 by selecting **Same as “Copy from end” document**. Use this option when you want to copy an existing trace relationship to a document to a branch of that document. This option is not available if you selected the **Same as “Copy to start” document** option in Step 3.
  - Create a trace relationship for a branch of the document you select in Step 4. The most recent predecessor is listed first and selected by default.

Click **Next**. The fourth panel of the **Propagate Traces** wizard displays.



- 7** Select the trace relationship(s) that you want to propagate to the document identified in Step 5. You can select any of the trace relationships that exist between the document identified in Step 3 and the document identified in Step 4. Usually there is only one relationship.

Click **Next**. The confirmation panel of the **Propagate Traces** wizard displays.



- 8** Click **Propagate**. A dialog box displays the status of the propagation.



---

# Importing and Exporting Documents

The MKS Integrity Client includes the **Document > Import** command for importing a document file into MKS Integrity and the **Document > Export** command for exporting a document to a file. The use of the commands and the format of the file are determined by factors outside of the scope of this documentation. For information on importing and exporting documents and supported file types, see the *MKS Gateway User Guide*.

## PART 8

# Executing Tests With MKS Integrity

---

# What is Test Management?

Testing is carried out using a test session item that identifies the test cases to be tested. Test cases can be contained in test suite documents or test execution group documents. You can create test session items and link them to test cases or documents, or you can create test sessions from test cases, test documents or test results. For more information on using test management, see the documentation included with the ALM solution download, available on the MKS Customer Community (<http://www.mks.com/community>).

For manual test sessions, you follow the documented instructions or test steps to execute the tests and enters the results. When entering results for a test you specify the verdict, for example, passed or failed. You can also enter notes for the test, and attach screen captures or other information related to the test. For failed tests, you can create a related defect.

For automated test sessions, you use an integrated third party testing tool to run the test cases. The results of the automated tests are sent back to MKS Integrity, where they can be viewed and edited. For more information on integrating with a third party testing tool, see MKS Customer Care.

# Creating a Test Session from Test Items

Testing is carried out using a test session item that identifies the test cases to be tested. You can create test session items and link them to test cases or documents, or you can create test sessions from test case items or test document items in the Items view, Relationships view, or Item Details view. Select the items you want to create the test session from and do the following:

- In the GUI, select **Test > Create Session From**
- In the Web UI, click  and select **Create Session From**

You can also create a test session from untested test case items. To display untested test cases, select a test session item (or any other test item that has the default Tests field visible), and do the following:

- In the GUI, select **Test > View Untested**
- In the Web UI, click  and select **View Untested**

Then use the **Create Session From** option to create the test session from the untested test cases.

---

**NOTE** You can also create a test session from test results, for example, test cases with a result verdict of Fail. You can create a test session from test results in the Test Results view or the **Test Results** tab on the Item Details view. For more information on the Test Results view, see “Test Results View: Analyzing Test Results” on page 233. For more information on the **Test Results** tab, see “Working With Item Test Results” on page 231.

---

# Viewing Test Result Details

---

**CLI EQUIVALENT** tm viewresult

---

The Test Result view lists the details of a test result, including any test steps.

To view the details of a test result from the Test Results view (GUI only):

- Select a test result and select **Test > View Result**.

To view the details of a test result from the Test Results tab on the Items Details view:

- In the GUI, select a test result and select **Test > View Result**.
- In the Web UI, click .

When viewing a test result in the GUI, you can click the value in the **Test Session** or **Test Case** field to display their item details.

MKS Integrity automatically provides an active e-mail link for the user specified in the **Modified By** field. Whenever an active e-mail hyperlink displays, you can send an e-mail to that user by clicking the link. If the user does not have an e-mail address, then no link displays.

If the test result has any relationships, attachments, or test steps a check mark displays on the respective tab. This allows you to see that the test result contains additional information without having to click the tab.

When viewing a test result in the GUI, select **View > Parameter Substitution** to display parameter values in text fields.

For information on creating or editing test results, see “Entering Test Results” on page 229.

## **Test Result Details Tabs**

The following information displays on the test result details tabs.

Tab	Description
<b>Test Result</b>	Displays the test result verdict and annotation, and fields from the test case. The test case fields that display are determined by your administrator. If the test case includes test steps, a summary of the test step results displays.
<b>Test Steps</b>	Displays the test steps in a relationship field. In the GUI, you can view test step details by right-clicking a test step.
<b>Related Items</b>	Displays related items for the test result, and related items for the test case that the test result belongs to.
<b>Attachments</b>	Displays list of attached files with the corresponding name, size, summary, date, and user details. You can open an attachment while viewing or editing a test result. To open an attachment in the GUI, double-click the attachment in the list or right-click the attachment field and choose <b>Open</b> . To open an attachment in the Web interface, click the attachment file name.

# Entering Test Results

To work with test results in the test result editor, the test session must be in a state that allows test results to be modified, and the test result policy for the session must allow you to modify test results. For more information on test session states and policies, contact your administrator. For more information on the test result editor, see “Working With the Test Result Editor” on page 234.

## To enter a test result

- 1 Select a test session item in the Items view or Relationships view, or open it in the Item Details view, and do the following:
  - In the GUI, select **Test > Result Editor**.
  - In the Web interface, click  and select **Test Result Editor**.
- 2 Select a test case in the tree pane of the Test Result editor.
- 3 On the **Test Result** tab in the details pane, do the following:
  - a) Select a select a verdict from the **Verdict** drop-down list.
  - b) Enter notes in the **Annotation** field. Notes should not exceed 4000 characters.
- 4 On the **Test Steps** tab, do the following:
  - a) Enter a verdict by selecting one or more test steps, right-clicking and selecting one of the following:
    - **Mark As Passed**
    - **Mark As Failed**
    - **Mark As >** (select other verdict)

---

### NOTE

- Test step verdicts do not affect the overall verdict for the test case.
  - You cannot add a related item or attachment to a test step result.
- 

- b) Enter notes by selecting a test step and doing the following:

- In the GUI, enter information in the **Annotation** column.
  - In the Web interface, double-click the **Annotation** column and enter information.
- 

**NOTE** In the Web interface, you can enter the same notes for multiple test steps by selecting the test steps, right-clicking and selecting **Set Annotation**.

---

- 5 On the **Related Items** tab, do the following:

- a) In the **Items related to this Test Result** field, add existing items or create new items as related items.
- 

**NOTE** When you create a new item as a related item, you can only create item types that your administrator has enabled to be related to test results.

---

- 
- b) In the **Items related to the results of Test Case <test case ID>** field, the related items are read-only and are intended to help you avoid creating duplicate defects. In the GUI, you can move a related item from this field to the **Test Result Related Items** field by selecting it and right-clicking, or using the drag-and-drop method.

---

**NOTE** Even when the test session is in a state that does not allow edits on the test results, you can still add or remove related items.

---

- 6 On the **Attachments** tab, add an attachment by clicking  next to the attachment field. Enter the path and name of the file to be attached, or browse to select a file. Click **Open**. The selected file is added to the attachments list.

In the GUI, you can add any image or text currently in the clipboard as an attachment by selecting **Test > Paste From Clipboard Attachment**. If an image is in the clipboard, it is added as `MKSImg<xxxx>.png`. If there is text in the clipboard, it is added as `MKSTxt<xxxx>.txt`. The attachment is added with the following summary: pasted attachment added automatically.

- 7 To save the test result, move to the next test case in the tree pane, or click **Test > Save Result** (GUI) or click  and select **Save Result** (Web).

---

**NOTE** To discard the test result, select **Test > Revert Result** (GUI) or click  and select **Revert Result** (Web).

---

# Working With Item Test Results

The **Test Results** tab displays the outcome of tests executed for an item. You can filter the results by test verdict (for example, passed, failed, or skipped) or test verdict type (pass, fail, or other). You can also filter to only see the last result recorded for the item. The total number of filtered test results displays at top right of tab.

The following columns display as icons if they are included in the **Test Results** tab:

Column	Column icon
Has Related Item	
Has Attachments	
Has Step Result	
Verdict Type Icon	
Verdict Icon	

You can also work with test results in the Edit Test Result view and the Test Result Editor. For more information, see “Editing Test Results” on page 237.

Test results are designed for use with the test management domain. For more information on test management, see documentation included with the ALM solution download, available on the MKS Customer Community (<http://www.mks.com/community>).

## Test Result Operations

The following table contains operations you can perform for test results on the Test Results tab when viewing item details:

To...	Do this...
View test result details	Select a test result and do one of the following: <ul style="list-style-type: none"><li>■ In the GUI, select <b>Test &gt; View Result</b>.</li><li>■ In the Web UI, click .</li></ul> The Test Result view displays. For more information, see “Viewing Test Result Details” on page 228.
Edit a test result	Select a test result and do one of the following: <ul style="list-style-type: none"><li>■ In the GUI, select <b>Test &gt; Edit Result</b>.</li><li>■ In the Web UI, click .</li></ul> The Test Result Editor displays. For more information, see “Working With the Test Result Editor” on page 234.
Clear a test result	Select a test result and do one of the following: <ul style="list-style-type: none"><li>■ In the GUI, select <b>Test &gt; Clear Result</b>.</li><li>■ In the Web UI, click .</li></ul>
View test case details	<ul style="list-style-type: none"><li>■ In the GUI, select a test result and select <b>Test &gt; View Test Case</b>.</li><li>■ In the Web UI, click the test case ID.</li></ul> <p><b>Note:</b> The test case information displays as of the test session’s Tests As Of Date, if one is specified for the session.</p>

---

To...	Do this...
View test session details (Web only)	In the Web UI, click the test session ID.
Create a test session based on selected test results	<p>Select the test results you want to create a test session for. For example, for a test session item you could select all the failed test results in order to re-test them. Then do the following:</p> <ul style="list-style-type: none"><li>■ In the GUI, select <b>Test &gt; Create Session From</b>.</li><li>■ In the Web UI, click </li></ul> <p><b>Note:</b> When you create a new test session from existing test results, the test case information is displayed as of the new session's Tests As Of Date. If this date is different than the original session's Tests As Of Date, the test case information may be different.</p>

# Test Results View: Analyzing Test Results

---

**CLI EQUIVALENT** `tm results`

---

The Test Results view displays a list of the test results for one or more test session items, test case items, or other item types that are related to test results (as set up by your administrator). This view is also embedded in the Item Details view for these item types. The Test Results view enables you to analyze test results (particularly failures) by providing a count of the results that meet the filter criteria, and enabling you to drill down to individual test case results data.

You can filter the results by verdict or verdict type. If you are viewing results for multiple test sessions, you can also filter to only see the last result that was recorded. The total number of filtered test results displays at top right of view.

---

**NOTE** If you edit a test result from this view so that it no longer satisfies the filter criteria, the test result is not cleared from the view until you select **View > Refresh**.

---

In the GUI, display the Test Results view by selecting one or more test session items in the Items view and selecting **Test > View Results**.

In the Web UI, test results are displayed on the Test Results tab of the Items Details view.

The following columns display as icons if they are included in the view:

Column	Icon
Has Related Item	
Has Attachments	
Has Step Result	
Verdict Type Icon	
Verdict Icon	

You can select a row in the view and perform the following operations.

Operation	Procedure
Edit the test result	<b>Test &gt; Edit Result.</b> Displays the Edit Test Result view. For more information, see “Viewing Test Result Details” on page 228.
View the test result details	<b>Test &gt; View Result.</b> Displays the Test Result view. For more information, see “Viewing Test Result Details” on page 228
Open the test session item for the test result	<b>Test &gt; View Test Session.</b> Displays the Item Details view for the test session.
Open the test case item for the test result	<b>Test &gt; View Test Case.</b> Displays the Item Details view for the test case.
Clear one or more test results	<b>Test &gt; Clear Results</b>
Create a new test session from one or more test results	<b>Test &gt; Create Session From</b>

# Working With the Test Result Editor

**CLI EQUIVALENT** tm resulteditor

The Test Result Editor enables you to quickly enter results for a manual test session. It displays the test cases for the test suites or test groups in the test session, including any test steps for the test cases.

If your administrator has integrated MKS Integrity with a third party testing tool, you can use the editor to review the results posted from the external tool. For more information on integrating with a third party testing tool, see the *MKS Integrity Server 2009 Installation and Configuration Guide*.

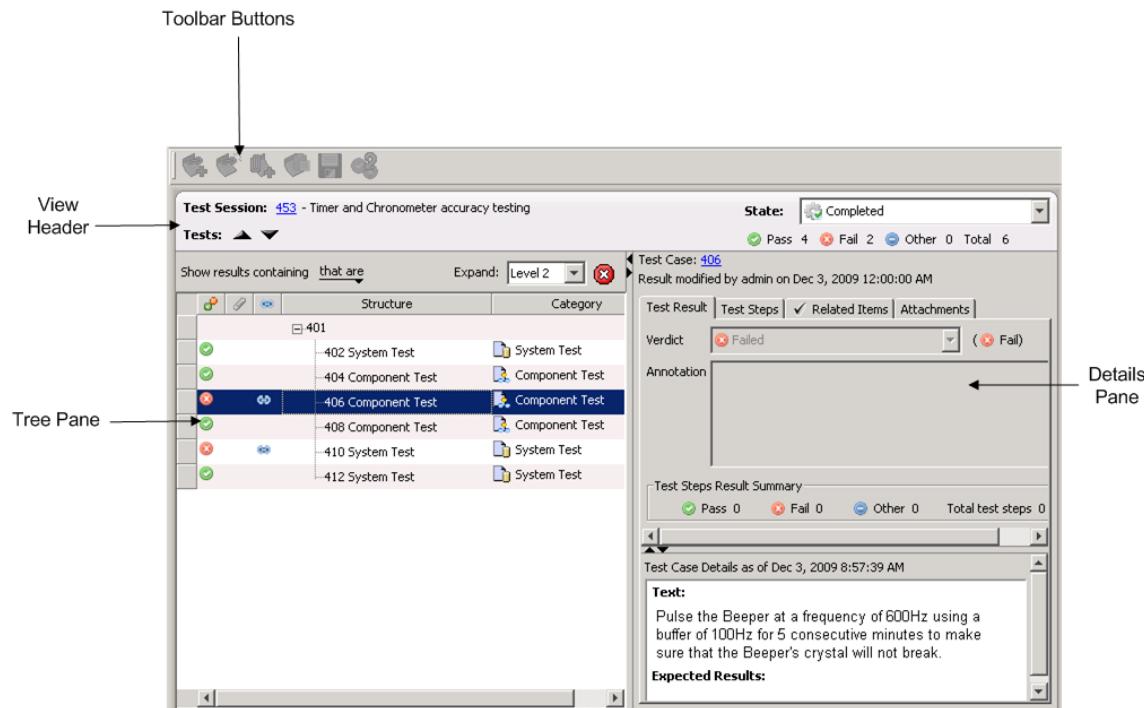
**NOTE** The test cases that display for test documents in the Test Result Editor are based on the date in the test session's **Tests As Of Date** field. Test cases added to test documents after this date are not displayed. If a Tests As Of Date date is not specified for the test session, the Test Result Editor displays test cases in the test document as of the current date.

To display the Test Result Editor for a test session item, select a test session item in the Items view or Relationships view, or open it in the Items Detail view, and do the following:

- In the GUI, select **Test > Result Editor**.
- In the Web interface, click  and select **Result Editor**.

## Test Result Editor: User Interface Components

In the GUI, the Test Result Editor view appears similar to the following:



The Test Result Editor view includes the following user interface components:

Component	Description
Toolbar Buttons (GUI only)	<p>Toolbar buttons enable you to perform the following actions without having to click the appropriate tab</p> <p> adds an existing related item to the test result. On the <b>Select Items</b> dialog box, do one of the following: Enter item IDs by typing the item numbers in the <b>Item IDs</b> field. To enter multiple IDs, separate each ID number with a space. Select from a list of available item IDs by clicking the browse button.</p> <p> creates a new item as a related item for the test result. <b>Note:</b> When you create a new item as a related item, you can only create item types that your administrator has enabled to be related to test results.</p> <p> adds any image or text currently in the clipboard as an attachment in the test result attachment field. If an image is in the clipboard, it is added as <code>MKSImg&lt;xxxx&gt;.png</code>. If there is text in the clipboard, it is added as <code>MKSTxt&lt;xxxx&gt;.txt</code>. The attachment is added with the following summary: <code>pasted attachment added automatically</code></p> <p> adds an attachment to the test result. Enter the path and name of the file to be attached, or browse to select a file. Click <b>Open</b>. The selected file is added to the attachments list for the test case.</p> <p> saves changes to the test result without moving to another row in the test result editor tree.</p> <p> discards unsaved changes to the test result.</p>
View Header	<p>Information about the test session is displayed at the top of the view, including the session state, verdict statistics for the session, and the total number of test cases in the session. You can change the state of a test session by selecting a different state in the <b>State</b> drop-down list. For example, you could change the state to <code>Complete</code> when you have entered results for all the test cases.</p>

Component	Description
Tree Pane	<p>The tree panel displays a structured list of the tests in a test session. The tree may have multiple root items, since the test session may have multiple test suites or test groups containing test cases. You can move up and down the structured list using the arrow buttons. Test results are saved automatically as you move through the tree.</p> <p>You can filter the results by verdict using the data filter at the top of the pane. You can filter results based on verdict or verdict type, including results with no verdict (unspecified).</p> <p>You can expand the tree to show the test cases contained in test suites, test groups, or test cases by clicking the plus icon.</p> <p>To collapse a branch of the tree, click the minus icon.</p> <p>In the GUI, the <b>Expand</b> pick list allows you to expand or collapse all nodes in the tree to a specific level. If you attempt to expand the tree, and the number of relationships is taking too long to display, click the  button to stop the expansion.</p> <p>In the GUI, you can drag-and-drop test items out of the tree but not into the tree. For example, if a test case failed you could create a defect, then drag-and-drop the failed test case to the appropriate relationship field on the defect.</p> <p>In the GUI, you can show (or hide) the tree pane and details pane by clicking .</p> <p>In the Web interface, you can:</p> <ul style="list-style-type: none"> <li>■ show (or hide) either pane by clicking </li> <li>■ view in list mode (click )</li> <li>■ view in detail mode (click )</li> <li>■ view in horizontal split mode (click )</li> </ul> <p>The following columns display as icons if they are included in the pane:</p> <ul style="list-style-type: none"> <li>■  indicates a verdict Verdict</li> <li>■  indicates has Related Item</li> <li>■  indicates has Attachments</li> </ul>
Details Pane	<p>The details pane displays the test result details view. For more information on using this view, see “Viewing Test Result Details” on page 228.</p> <p>If the test case currently highlighted is a test case that is a container for other test cases rather than containing test case content itself, the test result verdict and annotation information is not displayed and all the tabs for the test result details are read-only.</p> <p>In the GUI, you can click the value in the <b>Test Case</b> at the top of the view to display test case details. In the Web interface, you can display test case details by clicking  and selecting <b>View Item</b>.</p>

## Configuring the Test Result Editor

You can configure what fields display for item nodes in the tree pane of the test result editor.

In the GUI, select **View > Options**. On the **Titles** tab of the **Options** dialog box, use the **Fields** button to select the fields you want to display for each node. The field names are added with “{” and “}”.

In the Web interface, click  and select **Configure Options**. On the **Structure Format** tab of the **Configure View** dialog box, click  to select the fields that you want to display for each node. The field names are added with “{” and “}”.

# Editing Test Results

---

**CLI EQUIVALENT** tm editresult

---

To be able to edit a test result, the test session that the result belongs to must be in a state that allows test results to be modified, and the test result policy for the session must allow you to modify test results. For more information on test session states and policies, contact your administrator.

You can edit a test result in the Edit Test Result dialog box or the Test Result Editor.

To display the Edit Test Result dialog box:

- In the GUI, select a test result in the Test Results view or select a test result from the Test Results tab on the Items Details view and select **Test > Edit Result**.
- In the Web UI, select a test result from the Test Results tab on the Items Details view and click .

To display the Test Result Editor, select a test session item in the Items view or Relationships view, or open it in the Items Details view, and do the following:

- In the GUI, select **Test > Result Editor**.
- In the Web interface, click  and select **Test Result Editor**.

For more information on the Test Results Editor, see “Working With the Test Result Editor” on page 234.

---

**CLI EQUIVALENT** tm resulteditor

---

You can perform some operations in both the Edit Test Result view and the Test Result Editor.

## Editing Results in the Edit Test Result View

Operation	Procedure
Edit the test case results for a test case	Enter or change information on the result details tabs. For more information, see “Entering Test Results” on page 229.
Clear the verdict and notes for one or more test steps	On the <b>Test Steps</b> tab, select one or more test steps, right-click and select <b>Clear Result</b> .
Clear the verdict for one or more test steps	On the <b>Test Steps</b> tab, select one or more test steps, right-click and select <b>Clear Verdict</b> .

## Editing Results in the Test Result Editor

Operation	Procedure
Edit the test case results for a test case	In the details pane, enter or change information on the result details tabs. For more information, see “Entering Test Results” on page 229.
Batch edit test results for multiple test cases	In the tree pane, select multiple test cases or select a test suite or test group and do the following: <ul style="list-style-type: none"><li>■ In the GUI, select <b>Test &gt; Batch Edit Results</b>.</li><li>■ In the Web interface, click  and select <b>Batch Edit Results</b>.</li></ul> Check the <b>Verdict</b> option and select a verdict or check the <b>Annotation</b> option and enter a note. The information is applied to all the selected test cases or all the test cases in the selected suite or group. <b>Note:</b> You can clear the verdict for multiple test cases by checking the <b>Verdict</b> option and leaving it empty.

## Editing Results in the Test Result Editor

Operation	Procedure
Change the verdict for one or more test case results	<p>In the tree pane, select one or more test cases or select a test suite or test group, click <b>Test</b> (GUI) or  (Web) and select one of the following:</p> <ul style="list-style-type: none"> <li>■ <b>Mark As Passed</b></li> <li>■ <b>Mark As Failed</b></li> <li>■ <b>Mark As &gt;</b> (select other verdict)</li> </ul>
Add a related item to one or more test case results	<p>In the tree pane, select one or more test cases or select a test suite or test group and do the following:</p> <ul style="list-style-type: none"> <li>■ In the GUI, select <b>Test &gt; Add Related Item</b>.</li> <li>■ In the Web interface, click  and select <b>Add Related Item</b>.</li> </ul>
Add an attachment to one or more test case results	<p>In the tree pane, select one or more test cases or select a test suite or test group and do the following:</p> <ul style="list-style-type: none"> <li>■ In the GUI, select <b>Test &gt; Add Attachment</b>.</li> <li>■ In the Web interface, click  and select <b>Add Attachment</b>.</li> </ul>
Clear the verdict and notes for one or more test case results	<p>In the tree pane, select one or more test cases or select a test suite or test group and do the following:</p> <ul style="list-style-type: none"> <li>■ In the GUI, click <b>Test &gt; Clear Result</b>.</li> <li>■ In the Web interface, click  and select <b>Clear Result</b>.</li> </ul>
Clear the verdict for one or more test case results	<p>In the tree pane, select one or more test cases or select a test suite or test group and do the following:</p> <ul style="list-style-type: none"> <li>■ In the GUI, click <b>Test &gt; Clear Verdict</b>.</li> <li>■ In the Web interface, click  and select <b>Clear Verdict</b>.</li> </ul>
Clear the verdict and notes for one or more test steps	<p>In the details pane, on the <b>Test Steps</b> tab, select one or more test steps, right-click and select <b>Clear Step Results</b> (GUI) or <b>Clear Result</b> (Web).</p>
Clear the verdict for one or more test steps	<p>In the details pane, on the <b>Test Steps</b> tab, select one or more test steps, right-click and select <b>Clear Step Verdict</b> (GUI) or <b>Clear Verdict</b> (Web).</p>
Create a new test session from one or more test case results  For example, you could create a new test session from failed test results in order to re-test them.	<p>In the tree pane, select one or more test cases or select a test suite or test group and do the following:</p> <ul style="list-style-type: none"> <li>■ In the GUI, select <b>Test &gt; Create Session From</b>.</li> <li>■ In the Web interface, click  and select <b>Create Session From</b>.</li> </ul>

---

## Purging Test Results

Test results can be purged by your administrator using the `tm purgeresults` command. For more information on this command, see the *MKS Integrity 2009 CLI Reference Guide for Workflow and Documents*.

## PART 9

# Grouping Files Under Version Control

# Configuration Management Projects View

---

**CLI EQUIVALENT** `si projects`

---

From the **Projects** view in the GUI, you can open, create, import, and drop projects.

From the **Registered Projects** view in the Web interface, you can view the projects and project histories that are available on the MKS Integrity Server.

Operation	Procedure
To open the Projects view (GUI)	Select <b>Project &gt; View Projects</b>
To open the Registered Projects view (Web)	Select <b>Tools &gt; Manage Projects</b>

To display the menu of actions you can perform on a project in the GUI, select a project and right-click.

You can work with a subset of projects in the GUI by adding projects to the **My projects list** at the bottom of the view. To add projects to the **My projects list**, select a project from the projects list and click **+**.

The projects list automatically updates when subprojects are added, dropped, or reconfigured. However the **My projects list** does not automatically update to show such changes.

# Configuration Management Project Information View

**CLI EQUIVALENT** `si projectinfo`

The **Project Information** view displays the following information for a specified project:

- general information
- project attribute information
- development path information
- change package information
- associated MKS Integrity item information

Operation	Procedure
To view information for a project in the GUI	Select a project in the <b>Projects</b> view and select <b>Project &gt; Views &gt; View Information</b>
To view information for a project in the Web interface	Select a project in the <b>Registered Projects</b> view and select <b>Project &gt; Project Information</b>
To view information for a project checkpoint in the GUI and Web interface	Select a checkpoint in the <b>Project History</b> view, select <b>View Project</b> to open it, then select <b>Project &gt; Views &gt; View Information</b>

## General Project Information

The **General** tab displays the following project information:

Field	Description
<b>Project Name</b>	Displays the path and name of the project.
<b>Shared From</b>	Displays only if the project you selected is a shared or moved subproject, and displays the path and name of the project where the shared subproject originated.
<b>Server</b>	Displays the MKS Integrity Server name and port number where the project resides.
<b>Configuration Path</b>	Displays the configuration path of the project. The following keywords may appear in the path: # specifies a project or subproject in a well-formed project tree #p specifies a project that does not end with <code>project.pj</code> #s specifies a subproject in a poorly formed project tree #n specifies a normal project #d specifies the development path name #b specifies the number, label, or symbolic of the project checkpoint
<b>Type</b>	Type displays information describing the use of the project in Deploy. You only see this field if you are licensed to use Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .
<b>Members</b>	Number of members in the project, not including subprojects.
<b>Subprojects</b>	Number of subprojects in the project.

Field	Description
<b>Description</b>	Free-form text description of the master project. Edit the existing description, or type a new description.
<b>Last Checkpoint</b>	Revision number of last checkpoint, date and time of last checkpoint, and brief description of last checkpoint. You cannot change an existing checkpoint description from this dialog box, but you can append additional comments to it. To do so, enter any supplemental information in the <b>Checkpoint Description</b> field below the present information.

### Project Attributes

Configuration options in project files are sometimes called *attributes*. Within project files, you can set your own attributes as well as those pre-defined as options. You can set these attributes from the GUI or Web interface.

In the GUI, you can view and edit project attribute information through the **Attributes** tab. Attributes define local variables or set options. You can set master project attributes to apply across an entire project.

Attributes conform to the following formats:

- *variable*
- *variable*=*value*

---

**NOTE** No variable can exceed 80 characters, and no value can exceed 1024 characters.

---

For more information on project attributes, see the online help for the MKS Integrity Administration Client.

### Development Path Information

A development path is an identifier given to a new direction of project development. Changes made through the new development path are kept separate from the main development trunk unless you later choose to merge them.

In the GUI, the **Development Paths** tab shows a selectable list of any development paths created from the current project. A development path is identified by name, with the revision number (the development path is based on) in brackets, for example, Service\_pack2 (1.4).

---

**CAUTION** If you remove a development path that is referenced by a variant Sandbox, you can no longer open that variant Sandbox.

---

### Change Package Information

The **Change Package** tab displays the ID and summary of any change packages associated with subproject operations.

### Associated Item Information

The **Associated Items** tab displays the ID, type and summary of any MKS Integrity items associated with the project. Only item types that you have permission to view are displayed. You can right-click an associated item to view or edit it.

---

**NOTE** If you move a subproject that is associated with any MKS Integrity items, the items no longer display in this tab. You must open the MKS Integrity items and associate them with the subprojects in their new locations.

---

---

# Creating a Configuration Management Project

---

**CLI EQUIVALENT** `si createproject`

---

When creating a project, you only create the project container. To add files, or *members*, to the project, you must create a Sandbox on your client machine that points to the project and then add the project members through that Sandbox.

To create a project, select **Project > Create**.

Keep the following points in mind when creating a project:

- When you specify the name of the new configuration management project, MKS Integrity automatically assigns the .pj file extension. If you specify a .pj file extension in uppercase or mixed case, MKS Integrity replaces that file extension with the correct lowercase .pj file extension. If you specify a file extension other than .pj, MKS Integrity appends the .pj file extension to the file name.
- A single configuration management project name can be used only once in the same location.
- For case-insensitive repositories, MKS Integrity does not distinguish between configuration management project names differing only in case. For example, if `project.pj` already exists in `C:/Aurora_Program`, you cannot create `PROJECT.pj` in `C:/Aurora_Program`. This results in an error and MKS Integrity requires you to specify a different path and file name, or a different project name.
- Before creating reports you should be aware of the best practices for organizing reports.

---

# Importing a Configuration Management Project

---

## CLI EQUIVALENT `si importproject`

---

You can import configuration management projects if you need to do the following:

- restore projects that were dropped in earlier versions of MKS Integrity (including MKS Source and Source Integrity)
- migrate projects from Source Integrity Standard (an earlier version of MKS Integrity)

Importing a configuration management project registers it with the MKS Integrity Server. Once a project is imported, MKS Integrity commands can be performed upon it.

To import a project, select **Import Project**.

---

**NOTE** The Import Project command is not supported for database repositories. For information on how to import projects for database repositories, see the *Database Repository Migrator* document.

---

When importing projects, the import process automatically creates histories for any non-archived members and checkpoints any non-archived projects. Before you import a project, the project files must already reside in the server repository.

---

**CAUTION** MKS Integrity does not support encrypted archives. If your existing project has encrypted archives, you must first decrypt the archives before importing and registering the project with the MKS Integrity Server.

---

## **Importing From Earlier Versions**

If you are importing a configuration management project from an earlier version of MKS Integrity, once a project has been imported you may not perform commands on it using an older version of MKS Integrity. If for some reason you need to use an earlier version of MKS Integrity on a project, first drop the project using the Drop Project command. You must re-import the project before you can use it again with this version of MKS Integrity.

If you are importing an existing Source Integrity Standard project that includes out-of-tree members or subprojects, MKS Integrity automatically detects these and imports them into the newly registered project. A project member or subproject is considered out of tree when it does not reside in the project directory.

If you are importing a dropped project to its former repository location, the project must first be reclaimed by an administrator using the `si diag` command.

## **Key Considerations**

- If your connection to the MKS Integrity Server is disconnected while you are browsing for a file, the file browser does not show any files or directories.
- You cannot import a configuration management project that is already registered.
- For case insensitive repositories, MKS Integrity does not distinguish between configuration management project names that differ only in case. For example, if `project.pj` is already

---

registered in C:/Aurora\_Program, you cannot import PROJECT.pj into C:/Aurora\_Program. This results in an error and MKS Integrity does not import the project.

- The processing of this command may take some time, since it traverses all subprojects in the imported project and may perform various operations on them, such as checkpointing.
- An administrator may restrict the path names of configuration management projects being imported on the server. This is completed through the si.serverRoot property in the si.properties file on the MKS Integrity Server. If this is set, then any project location specified must be under that directory.

***Example***

The xyzBusiness company has an existing project in Source Integrity Standard. This project needs to be migrated to the new MKS Integrity Server. To carry out the import operation, the administrator first copies the project directory to the new host MKS Integrity Server and then imports it.

---

# Dropping a Configuration Management Project

---

**CLI EQUIVALENT** `si dropproject`

---

When a configuration management project has outlived its usefulness or just does not belong on the MKS Integrity Server anymore, you can remove it.

To *drop* a configuration management project means that the project is no longer registered with the MKS Integrity Server. Dropped projects can still be accessed as read-only from the **Change Package** and **Locks** views until the project is purged or reclaimed by your administrator.

To drop a configuration management project, select a project in the **Projects** view, and select **Drop Project**.

---

**CAUTION** If a project is dropped and any Sandboxes or variants are associated with it, those Sandboxes or variants no longer function.

---

Dropping a configuration management project unregisters it from the MKS Integrity Server, but it does not delete the project. Dropped configuration management projects can be added back into MKS Integrity.

---

# Adding a Configuration Management Project

---

**CLI EQUIVALENT** `si addproject`

---

You can add the following:

- a dropped project that still resides on the server (in the repository)
- an existing subproject as a top-level project

To add a project, click **Project > Add**.

If you need to restore a project from a backup or a dropped project from an older version of MKS Integrity (or MKS Source), use the Import Project command.

# Opening a Configuration Management Project

Opening a project in the GUI allows you to view the project and perform certain commands with its members.

Operation	Procedure
To open a project in the GUI	Select <b>Project &gt; Open Project</b> .
To open a project in the Web interface	Select <b>File &gt; Open Project</b> or Select <b>File &gt; Open Variant Project</b> or Select <b>File &gt; Open Build Project</b> The list displayed contains only master projects; subprojects are not accessible by this procedure.
To open a recently used project (GUI)	Select <b>Project &gt; Recent</b> , then select a project from the list.

When opening a project in the GUI, you can type in or select the project to open. If you are opening a variant or build subproject, initially only enter the path and name of the root project. You specify the subproject later in the procedure. When specifying the subproject, there are rules that control what project configuration you can jump to. If your selection breaks any of the rules, you cannot open the project.

When opening a project you can open the following types:

- **Normal** opens the working project based on the mainline.
- **Variant** opens a project based on a specific development path.

**NOTE** The **Variant** option is unavailable if there are no available development paths.

- **Build** opens a static project based on a specific checkpoint of the master project that is used for building or testing the project, but not for further development. You can specify the checkpoint through its checkpoint number or label.

## Alternate Methods of Opening a Project

- In the Web interface, if you know the location of a project, you can open it by typing the following URL in a browser:

`http://<server>:<port>/si/viewproject? projectName=<(sub)projectname>`

for example

`http://xyzBusiness:7001/si/viewproject? projectName=c:/master_projects/SourceCode/project.pj`.

- In the Web interface, you can also open a variant project from a **Project** view by selecting **Project > Open Variant Project**.
- In the GUI, you can open a project from the **Projects** view by selecting the project and clicking **Project > View Project**. You can open a build project from the **Project History** view by selecting a checkpoint and clicking **Project > View Project**.

---

## Configuration Management Project View

---

**CLI EQUIVALENT** `si viewproject`

---

When you open a configuration management project, MKS Integrity displays its contents in a **Project** view. The **Project** view displays subprojects and project members.

In the GUI, You can expand and collapse a project or subproject tree using one of the following options:

- Double-click the project or subproject.
- Click the plus or minus icons.
- Use the right and left arrow keys.
- Right-click the project or subproject, then select **Expand All**.
- Select **View > Expand All** or **View > Collapse All**.

When you select a member, information about the member displays in a box at the bottom of the view. For example, if the working file has been updated but not checked in, a message tells you that the working file has been changed.

In the Web, you can open a project or subproject by clicking the project or subproject link. You can also use the active links in the title bar to navigate within a project. For example, to navigate from a subproject to a project, click the applicable portion of the link.

---

### **Default Columns in Project View**

Column	Description
<b>Name</b>	Displays the name of the project, subproject, or member.
<b>Member Rev.</b>	Displays the member's revision number, or in the case of a subproject, the subproject revision number (if any), for example, 1.3.
	Indicates that your working file has changed. A delta symbol with a white document icon means your working file has changed. A delta symbol with a blue striped document icon indicates that the working revision does not match the member revision. A description of the changes is shown at the bottom of the window.
	A snowflake icon means the member is frozen. When the member is thawed, the snowflake icon disappears.
<b>Locked (Web only)</b>	Users who have a lock on the member. The padlock icon indicates the type of lock and any potential conflicts with other lockers. <b>Note:</b> Padlock icons in the Web interface are not color-coded. If there are multiple lockers, your lock is listed first, followed by the locker who has an exclusive lock (if any), followed by any lockers with non-exclusive locks. If there are multiple lockers, click on the field to display a drop-down list of lockers.
<b>Labels</b>	Displays any labels assigned to the member revision, for example, Draft1.
<b>State</b>	Displays the state of the member revision, for example, Beta.
<b>Member CPID</b>	Change package associated with the operation that set the member's revision. If you do not specify a change package when performing the operation, this column is not updated. In the Web interface, the change package ID displays as a hyperlink that you can click to display the change package. MKS Integrity displays a lock change package ID (a change package assigned during a checkout) in priority over the member revision change package ID (the change package assigned during a checkin).
<b>Symbolic Link</b>	In the Web, the symbolic link column indicates if the member is a symbolic link file. In the GUI, a symbolic link decorator  on the member icon indicates that the member is a symbolic link.

### **Filtering Members in the Project View**

You can view and work with a group of project members based on specific member properties. You can filter the members displayed using the **Filter** list located on the toolbar or select a group of members using **View > Select**.

# Creating a Configuration Management Subproject

---

**CLI EQUIVALENT** `si createsubproject`

---

MKS Integrity allows you to create large configuration management projects composed of smaller component projects. These smaller projects are known as *subprojects*. Subprojects behave in the same way as projects and can be accessed with most project and Sandbox commands.

For example, you could have subprojects for the following types of components:

- source files for creating individual executables
- source files for common libraries
- HTML files
- graphic files
- documentation

Each component usually has several related files. Each component can be kept in its own project. If you have two or more products that share these common components, you can create a separate master project for each product and include the projects for the common components as subprojects of each appropriate master project.

---

**NOTE** Subprojects are not the same as project members. You can only perform project-level operations on subprojects.

---

You create a new subproject, then add members and configure the subproject as necessary. For example, if you had a financial toolkit application that you wanted to add a new calculator application to, you would create a new subproject in the toolkit project to contain the calculator application source code.

To create a subproject in the GUI, select the project or Sandbox to create a subproject in, and select **Project > Subproject > Create**.

To maintain project integrity, the command may create empty subprojects as needed (that are visible when viewing the project revision history).

## Create Subproject Options

Column	Description
<b>Create one subproject per directory, as required</b>	Creates one subproject for each directory encountered when creating the subproject.
<b>If the project already exists, add it</b>	Adds an existing subproject to the project. This can be useful if the existing subproject had been dropped from a project.
<b>Project Name</b>	Specifies a name for the subproject, for example, <code>applications.pj</code> . MKS recommends using the <code>project.pj</code> default name.
<b>Close Change Package</b>	Closes the associated change package after the operation is complete.

---

# Adding a Configuration Management Subproject

---

---

**CLI EQUIVALENT** `si addsubproject`

---

You can add a subproject in the following two situations:

- A subproject was dropped from a project, the subproject's historical information remains, and you need to add the subproject back.
- You want to add a shared subproject.

To add a dropped subproject to a project in the GUI, select **Project > Subproject > Add**.

When adding a subproject, you can type in or select the subproject to add. MKS Integrity automatically determines if the subproject you specify is one that has been dropped. If the subproject already exists, you are presented with the option to share it.

When adding a subproject you can specify one of the following types:

- **Normal** adds a working subproject based on the mainline
- **Variant** adds a subproject based on a specific development path.

---

**NOTE** The **Variant** option is unavailable if there are no available development paths.

---

- **Build** adds a static subproject based on a specific checkpoint of the master project that is used for building or testing the project, but not for further development. You can specify the checkpoint through its checkpoint number or label.
- **Default** adds a subproject as the same type as the parent project.

---

**NOTE** If you do not select a subproject type, the subproject is added as the same type as the parent project.

---

---

# Adding a Shared Configuration Management Subproject

---

**CLI EQUIVALENT** `si sharesubproject`

---

A *shared subproject* is a subproject that is a member of more than one configuration management project. You can share a subproject between two or more projects by referencing the original subproject. A shared subproject allows you to access common members across many projects. Shared subprojects are not required to be located within the same directory structure or project hierarchy.

For example, Ryan maintains the North American and German Web sites for ABC Financial. Both Web sites are under version control and use the same images, but they contain different content. To avoid duplication and ensure both Web sites contain the same images, Ryan adds the images to a subproject and shares it between the two projects.

To add a shared subproject, select the project or Sandbox you want to add the shared subproject to, and select **Project > Subproject > Add Shared**.

When adding a subproject you can specify one of the following types:

- **Normal** adds a subproject based on the working subproject on the mainline
- **Variant** adds a subproject based on a specific development path of the master project.

---

**NOTE** The **Variant** option is unavailable if there are no available development paths.

---

- **Build** adds a static subproject based on a specific checkpoint of the master project that is used for building or testing the project, but not for further development. You can specify the checkpoint through its checkpoint number or label.
- **Default** adds a subproject as the same type as the parent project.

---

**NOTE** If you do not select a subproject type, the subproject is added as the same type as the parent project.

---

When you type the destination subdirectory for the shared subproject, the subproject name must reside in the project directory.

## **Shared Project Behavior**

A shared subproject functions the same as an unshared subproject and is accessible by the same commands. The shared subproject continues to reside within its original master project, but is referenced by the other project and shown as a shared subproject.

When working with shared subprojects, MKS Integrity uses the actual name of the subproject in the repository rather than its relative name in the project hierarchy for the purposes of resolving ACLs, policy statements, event triggers, and change package entries. This enhances the portability of change packages across different projects.

## **Shared Projects From Source Integrity Standard**

Shared subprojects (out-of-tree subprojects) that were created in Source Integrity Standard (an earlier version of MKS Integrity) are detected as they are accessed by MKS Integrity without disrupting the

---

operation. The format of these subprojects is retained until you change or update it to the new format by reconfiguring it.

# Configuring a Configuration Management Subproject

---

**CLI EQUIVALENT** `si configuresubproject`

---

Once you create or add a subproject, you can modify the type to suit your needs. For example, you can change a normal subproject to a build subproject to suspend development, or you can change a variant subproject to a normal subproject to continue development on the main trunk.

For example, Jen wants the ABC Tools development team to work with revision 1.2 of `amortization/project.pj`, the last known stable version of the code. Jen does not want the general development team to do any further development on this release, so she configures the `amortization/project.pj` subproject to be a build subproject.

Interface	Procedure
GUI	Select <b>Configure Subproject</b> .
Web	Select <b>Project &gt; Configure Subproject</b> .

When configuring a subproject you can specify one of the following types:

- **Normal** configures a subproject to the working subproject on the mainline.
- **Variant** configures a subproject to a specific development path.

---

**NOTE** The **Variant** option is unavailable if there are no available development paths.

---

- **Build** configures a static subproject to a specific checkpoint of the project that is used for building or testing the project, but not for further development. You can specify the checkpoint through its checkpoint number or label.

## **Changes Between Configurations**

Any changes you make when configuring a subproject affects the project as a whole and any shared subprojects within it. Deltas reflecting these changes appear in the Project or Sandbox view. Some differences you may see include:

- Members existing in the original version and configured version of the subproject display a delta if the working revision in the Sandbox is different from the new member revision.
- Members that did not exist in the original version of the subproject, but do exist in the configured subproject, display a delta to indicate that the Sandbox does not have a working file for the new member.
- Members that existed in the original version of the subproject, but do not exist in the configured subproject, display as former members.
- Subprojects that existed as members in the original version of the subproject, but do not exist in the configured subproject, display as former subprojects.

To resolve the differences, resynchronize the subSandbox.

## **Configuring Shared Subprojects**

When configuring shared subprojects, remember that each shared subproject is configured independently. This means that when you configure a shared subproject, the reconfiguration does not

---

change all instances of that subproject. For example, if the subproject `tools/project.pj` is shared in the two projects, `Aurora/project.pj` and `Libra/project.pj`, a change to the configuration of `Aurora/tools/project.pj` does not affect the configuration of `Libra/tools/project.pj`.

### ***Source Integrity Standard Subprojects***

Configured subprojects or frozen subprojects created in Source Integrity Standard (an earlier version of MKS Integrity) are detected as they are accessed by MKS Integrity without disrupting the operation. The format of these subprojects is retained until you change or update it to the new format by reconfiguring it.

---

# Dropping a Configuration Management Subproject

---

**CLI EQUIVALENT** `si dropproject`

---

If a subproject has outlived its usefulness or just does not belong in a project anymore, you can remove it at any time.

To drop a subproject, select one or more subprojects to remove and select **Project > Subproject > Drop**.

After you remove a subproject from a project, it is no longer listed as part of the Sandbox or master project, but the subproject's history remains in the project record, in case you need to recreate an earlier version of the project.

---

**IMPORTANT** When you remove a subproject, you also remove all members within that subproject.

---

---

# Moving a Configuration Management Subproject

---

**CLI EQUIVALENT** `si movesubproject`

---

To meet the needs of changing project configurations, you can move one or more subprojects and all of its members and sub-subprojects between projects, and/or directories in a single project, or variants of the same project on the same MKS Integrity Server.

For example, the size of the documentation set for the ABC Financial toolkit has increased significantly since the last release. To reduce the size of the toolkit installer, product management wants the documentation available from the CD browser only. Michelle moves the `documentation.pj` subproject from the `toolkit.pj` project to the `cd_browser.pj` project.

When a subproject is moved, it behaves like a shared subproject. The subproject in the new location continues to be backed by the underlying subproject in the old location, and the path and name of the subproject file in the repository remains the same. Any external references (ACL names, event triggers, policy statements) to the moved subproject continue to work because they are based on the subproject's original name; however, a subproject that is moved into a new project hierarchy continues to inherit ACLs from its original hierarchy and not from the new parent project. The moved subproject also retains its configuration type (normal, variant, build). If you are moving multiple subprojects, any common directory prefix shared by the subprojects is automatically removed during the move.

You can move a subproject through a **Project** or **Sandbox** view using the **Project > Subproject > Move** command.

You can also move a subproject by dragging it onto a project, Sandbox, subproject, sub Sandbox, or directory node in the active **Project** or **Sandbox** view, or onto an adjacent open **Project** or **Sandbox** view. The drag-and-drop action initiates the **Move Subproject Wizard**, summarizing the details of the move.

## **Key Considerations**

- Moving subprojects between projects on different servers is not supported.
- The moved subproject inherits the project or directory ACLs from its original location. You cannot apply the ACLs from the new location to the subproject.
- The path and name of the subproject file in the repository is permanently reserved. If you attempt to create a new subproject using the moved subproject's original path and name in the repository, you are prompted to add the existing subproject. If you answer no, the create subproject operation exits without providing you with the option of creating a subproject with a different path and name.
- Deferred subproject moves are not supported.
- The `AddSubproject` permission is required on the target project, and the `DropSubproject` permission is required on the source project. The `ConfigureSubproject` permission is required for moving heterogeneous subprojects.
- You can move one or more subprojects across directories within a single project.
- Move subprojects do not display as shared in a **Sandbox** or **Project** view unless the subproject was shared before the move.

- When you move one or more subprojects, you cannot co-locate subprojects in the same directory. If you want to co-locate an existing subproject with another subproject, perform an Add Shared Subproject operation, followed by a Drop Subproject operation.
- If you move a subproject that is associated with any MKS Integrity items, the items no longer display on the **Associated Items** tab for the project. You must open the MKS Integrity items and associate them with the subprojects in their new locations.
- If you are typing in or selecting a destination project or Sandbox instead of using the default, and you are moving the subproject to a variant or build subproject, initially only enter the path and name of the root project. You specify the subproject later in the procedure. When specifying the subproject, there are rules that control what project configuration you can jump to. If your selection breaks any of the rules, you cannot move the subproject.

## Move Subprojects Options

Options	Description
<b>Move Working Files</b>	Controls whether to move existing working files in the subproject you are moving. <ul style="list-style-type: none"> <li>▪ <b>None</b> does not move any existing working files.</li> <li>▪ <b>Subsandbox members only</b> moves only the existing working files in the sub Sandbox.</li> <li>▪ <b>Entire Subsandbox Directory</b> moves the entire sub Sandbox directory.</li> </ul> <p><b>Note:</b> This option is valid only if you are moving one or more subprojects from a source Sandbox to a target Sandbox.</p>
<b>Confirm Move</b>	Confirms the move before proceeding.
<b>Create Subprojects</b>	Creates subprojects in directories that do not contain subprojects.
<b>Overwrite Existing Files</b>	Overwrites working files if they exist in the new location. <p><b>Note:</b> This option is valid only if you are moving one or more subprojects from a source Sandbox to a target Sandbox</p>
<b>Close Change Package</b>	Closes the associated changed package.

# Viewing a Configuration Management Project History

**CLI EQUIVALENT** `si viewprojecthistory`

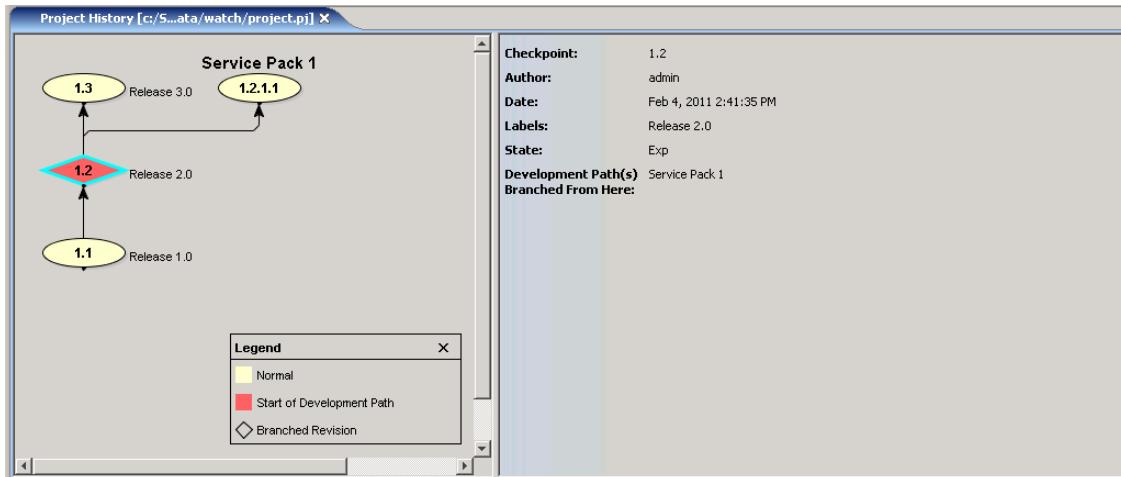
The **Project History** view is a window that displays the history of a project, including the checkpoint number and description, author, date, labels, and promotion state of the project.

Interface	Procedure
GUI	Select <b>Project &gt; Views &gt; View History</b>
Web	Select <b>Project &gt; View Project History</b>

By default, the project history view displays two panels: a panel showing the path of development, and a panel showing details for a selected checkpoint. In order for information to display in the details panel, you must select a checkpoint in the GUI or click a revision number in the **Revision** column in the Web interface.

In the GUI, you can use a project history filter to display any checkpoints of the current project that meet specified selection criteria. You can change the filter by selecting **View > Change Filter**.

In the GUI, you can toggle between a graphical view and a list view by clicking **View > List** or **View > Graphical**.



*Graphical project history view*

The information in the graphical view shows the path of development from checkpoint to checkpoint, including branches and merge lines. You can change the information that displays for each checkpoint by selecting **View > Show beside each node**. If summary information is not displayed in the view, it displays in a tooltip when you place your mouse pointer on a specific checkpoint.

Project History [c:/S...ata/watch/project.pj] X				
Revision	Author	Date	Labels	
1.3	admin	Feb 4, 2011 2:46:04 PM	[Release 3.0]	
1.2.1.1	jriley	Mar 7, 2011 3:30:51 PM		
1.2	admin	Feb 4, 2011 2:41:35 PM	[Release 2.0]	
1.1	admin	Feb 4, 2011 2:41:18 PM	[Release 1.0]	

**Checkpoint:** 1.2  
**Author:** admin  
**Date:** Feb 4, 2011 2:41:35 PM  
**Labels:** Release 2.0  
**State:** Exp  
**Development Path(s)** Service Pack 1  
**Branched From Here:**

### List Project History view

The information in the list view is displayed in columns. If the **Associated Items** column is displayed, and there are multiple MKS Integrity item associated with the checkpoint, clicking on a cell in this column changes it to a drop down list.

The details panel in the graphical or list view displays additional details for a selected checkpoint. You can turn the display of this panel on or off using the **View > Show Details** option. The following information can display in the project history details.

Field	Description
<b>Checkpoint</b>	Checkpoint number of the project
<b>Author</b>	User who checkpointer the project
<b>Date</b>	Date the checkpoint was created
<b>Labels</b>	Label for checkpoint.
<b>State</b>	State of the project
<b>Checkpoint Description</b>	Description for checkpoint
<b>Development Path</b>	Development path of checkpoint
<b>Development Path(s) Branched From Here</b>	Any development paths that branch directly from the checkpoint.

If you have configured the **Project History** view to be dynamic, it is automatically updated to reflect changes in the **Projects** view and the **Project** view. If you are using MKS Deploy, this view is also automatically updated to reflect changes in the **Staging System** view, **Stage** view, and the **Staging Systems** view. For more information on MKS Deploy, see the *MKS Deploy 2009 Administration Guide*.

# Restoring a Configuration Management Project

---

**CLI EQUIVALENT** `si restoreproject`

---

The Restore Project command allows you to restore a configuration management project to a previously checkpointed revision. Restoring a project is useful when development must revert back to an earlier version and there are no plans to proceed from the current version of the project. Any further development then proceeds from the restored project revision. The Restore Project command can be applied to both normal and variant projects.

---

**NOTE** The Restore Project command can potentially restore and checkpoint dropped subprojects that existed at the target revision, even if they are not currently a member of the project.

---

To restore a project through the GUI, select the configuration management project that you want to restore in either a Project or Sandbox view, and select **Restore Project**.

---

**NOTE** When you work through a Sandbox or sub Sandbox, the corresponding master project is referenced.

---

---

**IMPORTANT** Do not use the **Restore Project** command to create a *new* development branch from a previously checkpointed project. For new development paths, you should instead create a development path.

---

## How the Restore Project Command Works

MKS Integrity performs the Restore Project command as follows:

- A checkpoint is performed on the current configuration management project revision.
- The configuration management project is restored to the target revision.
- A final checkpoint of the restored revision is made.

Therefore, for each configuration management project you restore, two revisions are generated. For example, if the head revision of the project is 1.4 and you decide to restore it to revision 1.2, the following project revisions are generated:

- 1.6 final checkpoint
- 1.5 pre-checkpoint

You would then continue your project development work from revision 1.6.

## Selecting a Checkpoint to Restore

You can select the checkpoint to restore by selecting a **Pre-Defined Revision** or a **Specific Revision**.

If you want to restore a pre-defined revision, select one of the following:

- **Head** revision, which represents the latest checkpoint on the default branch of development (or on the mainline, if no default is specified)
- **Trunk Tip**, which represents the latest checkpoint on the mainline independent of the default branch settings.

---

If you want to restore a specific revision, you can select the revision based on a checkpoint number or label by clicking the appropriate tab. The default revision is the most recent checkpoint.

### **Key Considerations**

- When a configuration management project is restored, all restored members return to the initial state.
- The **Restore Project** command can be applied to both normal and variant projects.
- You can effectively undo the **Restore Project** command by restoring the configuration management project to the pre-checkpointed revision.
- You cannot restore a build project using the **Restore Project** command.
- You cannot restore a project that is associated with a deploy stage. For more information on deploy stages, see the *MKS Deploy 2009 Administration Guide*.
- You cannot restore a project if a checkpoint is in progress on that project.
- To restore a variant project to a specific project revision, the development path must exist in all subprojects referenced by the project revision.

---

# Adding Configuration Management Project Labels

You can add labels to project checkpoints. *Labels* are unique text that describe the product stage or release, or the content of the project. Project labels can be based on any information that would be useful in identifying that particular checkpoint of the project, for example, a code freeze. If there is more than one label on a project, the labels display in alphabetical order.

Project labels can also be added as part of the checkpoint process.

Labels cannot contain colons (:), square brackets ([ ]), leading spaces, numbers in the same format as a valid revision number (1.23), or numbers without any spaces (14325). Numbers that contain spaces are acceptable, for example, 2432 1234.

Operation	Procedure
To add a project label in the GUI	Select a checkpoint in the <b>Project History</b> view and select <b>Project &gt; Properties &gt; Add Label</b>
To add a project label in the Web interface	Select a checkpoint in the <b>Project History</b> view and click <b>History &gt; Properties &gt; Add Label</b> or Select a project in the <b>Project</b> view and select <b>Project &gt; Add Label</b>

---

**NOTE** You cannot add a label to a project if a checkpoint is in progress on that project.

---

In the GUI, to move an existing label to another checkpoint, click **Options**, then select the **Move Existing Label** option.

---

# Deleting Configuration Management Project Labels

---

**CLI EQUIVALENT** `si deleteprojectlabel`

---

You can delete project labels. Deleting project labels may be appropriate in situations where a label is no longer relevant or where an incorrect label is applied.

Operation	Procedure
To delete a project label in the GUI	Select a project or Sandbox and select <b>Project &gt; Properties &gt; Delete Label</b> . In the <b>Delete Label</b> dialog box, select a label to delete from the <b>Label</b> list.
To delete a project label in the Web interface	Select a project and select <b>Project &gt; Delete Label</b> or <b>History &gt; Delete Label</b> . In the <b>Delete Label</b> dialog box, select a label to delete from the <b>Label</b> list.

---

**NOTE** You cannot delete a label from a project if a checkpoint is in progress on that project.

---

# Checkpointing a Configuration Management Project

---

**CLI EQUIVALENT** `si checkpoint`

---

*Checkpointing* a project creates a new revision of the project and adds it to the project history. When you checkpoint a project, you save all the information needed to recreate the project completely as it existed when you checkpointed it. The saved information includes the project structure and the list of members with their revision numbers.

For example, Release 3.0 of the ABC Financial Toolkit is approaching and work on the amortization project has been completed and passed to Quality Assurance. To preserve the project structure, Steve checkpoints the amortization project.

When you checkpoint a project in MKS Integrity, it creates a revision of a project in the project history. You can view the history of a project, restore a project to an earlier checkpoint, and compare differences between checkpoints.

If you want to create a variant or build Sandbox, you must first checkpoint the project.

If checkpointing a variant project (development path), the variant of the project must be displayed and selected prior to initiating the checkpoint operation.

Interface	Procedure
GUI	Select a project or Sandbox and select <b>Project &gt; Checkpoint</b>
Web interface	Select a project and select <b>Project &gt; Checkpoint Project</b>

## Checkpointing With a Database Repository

If you are using a database type repository, all checkpoints are *transactional*. This means that a checkpoint records the structure and contents of the project at the time the checkpoint starts. It does not include anything added to the project or its subprojects after the checkpoint starts, such as checkins or submitted change packages.

---

**NOTE** To ensure that only complete change packages are included in your project checkpoints, your administrator must enable change package reviews.

---

While the checkpoint is in progress, you can perform member and subproject operations on the project, but you cannot perform any operations that affect project checkpoints, development paths, or labels. You cannot perform the following operations:

- Checkpoint the project on another development path
- Restore the project to a previous checkpoint
- Delete the project from the database
- Add or delete project labels
- Create or remove development paths

For more information on deleting projects, see the online help for the MKS Integrity Administration Client.

---

### **Key Considerations**

- Checkpointing a project affects the project only; it does not check in every member of the project.
- If you are working in a regular Sandbox, issuing a checkpoint command checkpoints the Sandbox's master project.
- Checkpointing a project checkpoints all its subprojects. If you do not want to checkpoint subprojects, configure them as build subprojects first.
- You can use the project's revision number to keep track of your projects, but to simplify post-release maintenance, use a label to identify significant project development milestones when you checkpoint a project. A checkpoint label is a unique text string assigned by you to identify a new project checkpoint, for example, Beta. Labels cannot contain colons (:), square brackets ([ ]), or leading spaces. Additionally they cannot have the same format as a valid revision number.

---

**NOTE** If you specify a label that is the same as one used for another checkpoint in the history and have the `MoveProjectLabel` permission, the label from the earlier checkpoint is moved to the new checkpoint. For more information on permissions, contact your administrator.

- Using the **Apply Label to All Members** or **Apply State to All Members** options when checkpointing a project slows down the checkpoint operation considerably. Do not select these options unless there is a definite need to label or set the state of all members individually.

# Comparing Configuration Management Project Checkpoints

As you work with the project and its members, the content of the project file changes:

- The revision number of the project is updated each time you checkpoint that project.
- As you check in individual project members, their revision numbers change.
- Project members may be added or deleted.
- Project attributes may be added or deleted.

The **View Project Differences** command can compare the following:

- two specified checkpoints of the project
- working project (or the last checkpoint of a build project's branch) with a specified checkpoint
- working project (for normal or variant projects) with the last checkpoint

**NOTE** The **View Project Differences** command only compares project checkpoints that are within a single line of development; it does not compare project checkpoints across branches.

Information is provided on any change packages that have been applied between two checkpoints of the project (or one checkpoint and the current version). This information is extremely useful for confirming the specific content of a project.

Operation	Procedure
Compare two checkpoints	From the <b>Project History View</b> , select the two checkpoints of the project you want to compare and click <b>Project &gt; Views &gt; View Differences</b> (GUI) or <b>History &gt; View Project Difference</b> (Web).
Compare the working project with a checkpoint	From the <b>Project History View</b> , select a checkpoint and click <b>Project &gt; Views &gt; View Differences</b> (GUI) or <b>History &gt; View Project Difference</b> (Web).
Compare the working project with the last checkpoint	From the <b>Project view</b> , click <b>Project &gt; Views &gt; View Differences</b> (GUI) or <b>Project &gt; View Project Differences</b> (Web). Note: For normal projects, the last checkpoint is the last checkpoint on the trunk; for variant projects, it is the last checkpoint on the variant's branch. For build projects, MKS Integrity compares the contents of the build project checkpoint with the last checkpoint on the build project's branch.

## Configuration Management Project Differences View

The **Project Differences** view displays project differences between checkpoints of project. The changes to a project and its members are not tracked as they happen, but rather the **View Project Differences** command takes two checkpoints (or one checkpoint and the current version) and then compares them.

### **By Member Panel**

The **By Member** panel displays the following information:

- **Name** displays name of the project, subproject, or member changed.
- **Change** displays the operation performed on the member or subproject, and the resulting revision number.

---

Selecting a member entry displays the following information:

- **Revision** displays the member's revision number.
- **Author** displays the author of the revision.
- **Date** displays the date each revision in the history was created.
- **Labels** displays revision labels.
- **State** displays the state of the member revision.
- **C.P. ID** displays the revision's associated change package ID.

---

**NOTE** Members that were checked in between the two checkpoints without updating the revision number do not appear in the **Project Differences** view, and do not display change package information in the **By Change Package** panel.

---

### **By Change Package Panel**

Change package information displayed on the **By Change Package** panel is drawn from the modifications calculated from the checkpoint comparison.

The **By Change Package** panel displays the following information:

- **ID** displays the change package ID for change packages that contain entries with member changes between project checkpoints.
- **Summary** displays the change package summary for change packages that contain entries with member changes between project checkpoints.

Modifications that are not associated with a change package (unresolved) are displayed in the bottom frame of the **By Change Package** panel. Where possible, the member or project name displays with full path information.

---

**NOTE** The **By Change Package** panel does not show which change packages have been propagated between project checkpoints on different branches.

---

The following are reasons why modifications can appear in the bottom frame:

- The change package entry is missing, either because one was not recorded for the member operation or the entry was discarded.
- The operation that made the modification cannot use a change package, for example, adding a subproject.
- The project state captures compared are from different development paths.

Pending change package entries do not appear in the **By Change Package** panel.

# Viewing Configuration Management Project Metrics

---

**CLI EQUIVALENT** `si addprojectmetric`

---

If metrics are being tracked for MKS Integrity configuration management projects, you can view the calculated metrics for a specific project checkpoint. For information on setting up metrics for projects, see the online help for the MKS Integrity Administration Client.

Operation	Procedure
To view project metrics for a build project in the GUI	Open a build project and click <b>Project &gt; View &gt; View Metrics</b>
To view project metrics for a normal or variant project in the GUI	Select a revision in the <b>Project History</b> view and click <b>Views &gt; View Metrics</b>
To view project metrics for a build project in the Web interface	Open a build project and click <b>Project &gt; View Project Metrics</b>
To view project metrics for a normal or variant project in the Web interface	Select a revision in the <b>Project History</b> view and click <b>History &gt; View Project Metrics</b>

The **Metrics** view displays the calculated value for each metric, the number of files used in the metric calculation, and the average value of the metric across all files used in the metric calculation.

Metric Name	Description	Value	Count	Average
lines	Physical lines	305	29	10.517
characters	Number of characters in text files	6981	29	240.724
bytes	Number of bytes in binary files	5404	16	337.75
revisions	Number of revisions in archive	48	45	1.067
members	Number of normal members	45		
subs	Number of subproject members	0		
checkpoints	Number of checkpoints	5		
javafiles	Number of java source files	17		
binary	Number of binary files	16		
text	Number of text files	29		
cfiles	Number of C or CPP files	9		
maxmembers	Largest number of members in any subproject	45		
maxsubproj...	Largest number of subprojects in any subproject	0		

---

**NOTE** Metrics are calculated automatically by event triggers. You can calculate metrics manually in the **Project History** view of the GUI by selecting a project checkpoint and clicking **Project > Calculate Project Metrics**.

---

The **Metrics** view displays the following information:

Column	Description
<b>Metric Name</b>	Name of the metric.
<b>Description</b>	Description of the metric.
<b>Value</b>	Calculated value of the metric.
<b>Count</b>	Number of files used in the metric calculation.
<b>Average</b>	Average value of the metric across all files used in the metric calculation.



---

# Branching Configuration Management Projects Overview

Branching projects is useful for:

- extracting and building from previous versions of a project
- building customized versions of a product
- performing branch development work
- performing post-release maintenance
- fixing defects in previous versions of the product
- testing new features outside of the main development path
- experimenting with research that does not affect regular development

You create a project branch by creating a new development path.

There are main models to follow when branching projects:

- release based branching
- project based branching

## ***Release Based Branching***

In a release based branching model, the content of the project (features, enhancements, and defects) tends to be static and well defined. Typically, the complete project goes through the stages of development, testing, and release.

All development activity is performed on the main trunk of the project. Release candidates are identified as checkpoints and finalized on a project branch. Bug fixes and stabilization development work continues on the main trunk of the project and are migrated to the release candidate branch using Apply Change Package.

This type of branching model is suited for development environments where full versions are released into production. When using this model, it is recommended to limit the number of branches to avoid the need for mass merges at a later point.

## ***Project Based Branching***

In a project based branching model, the content of the project (features, enhancements, and defects) tends to be dynamic, with constant additions and removals of subprojects (comprised of features, enhancements, and defects) until the last minute. The project is released in increments into production. Usually, this type of development does not release the entire project after the initial rollout, and tends to have updates to production as frequently as every week.

All development is performed on project branches created from the latest production checkpoint. Changes made on project branches are merged into an integration branch using the resynchronize change package command and tested on that branch before being merged into the main trunk. After newly introduced integration code has been released on the main trunk, all active project branches are dropped and re-created from the latest main trunk (production) checkpoint. This forces all active project branches to synchronize with the latest production release.

In a project based branching model, it is recommended that you only create branches for concurrent or parallel development projects. Creating branches at the feature, enhancement, defect or task level is not recommended.

# Creating a Development Path

---

**CLI EQUIVALENT** `si createdevpath`

---

A *development path* is an identifier given to a new branch of software development. Changes made through the new development path are kept separate from the main development trunk unless you choose to merge them later.

For example, a user group at ABC Financial requests a version of the Stock Calculator that has the commission fields removed and a special commissions legal message added. The task is assigned to Chad. He must use version 1.0 of the stock project that did not contain the commission fields and then add the new legal message to that version. To do this, he creates a development path and a variant Sandbox off of the checkpointed 1.0 version.

Once you have created a development path, you can open and work in a variant project by opening a variant Sandbox.

MKS Integrity allows multiple developers to point to the same development path, each using their own variant Sandbox. In the variant Sandbox, you can see the current state of the project along the development path and the changes made by other developers using it.

When a development path is created for a project, it is also created for all subprojects, reserving the assigned name as a unique identifier and ensuring no two paths can share the same name. The project is also automatically checkpointed when a development path is created.

Interface	Procedure
GUI	Select a project or Sandbox and select <b>Project &gt; Development Path &gt; Create</b> .
Web	Select a project and select <b>Project &gt; Create Development Path</b> or <b>History &gt; Create Development Path</b> .

---

**NOTE** You cannot create a development path for a project if a checkpoint is in progress on that project.

---

## Selecting the Checkpoint to Create a Development Path From

You can select the checkpoint to create a development path from by selecting a **Pre-Defined Revision** or a **Specific Revision**.

If you want to create a development path from a pre-defined checkpoint, you can select one of the following:

- the `Head` revision, which represents the latest checkpoint on the default branch of development (or on the mainline, if no default is specified)
- the `Trunk Tip`, which represents the latest checkpoint on the mainline independent of the default branch settings.

If you want to create a development path from a specific checkpoint, you can select the checkpoint based on a checkpoint number or label by clicking the appropriate tab. The default checkpoint is the most recent checkpoint.

You can view the development path of a checkpoint, plus any development paths branched from a checkpoint, by selecting the checkpoint in the **Project History** view and viewing the details panel.

# Removing a Development Path

Once a development path is no longer needed or useful, you can remove it at any time. Deleting a development path is permanent; however, any changes you checked in through the development path remain in your repository.

**CAUTION** If you remove a development path that is referenced by a variant Sandbox, you cannot open that variant Sandbox.

Interface	Procedure
GUI	Select a project or Sandbox and select <b>Project &gt; Development Path &gt; Remove</b> . In the <b>Remove Development Path</b> dialog box, select a development path to remove from the <b>Development Path</b> list.
Web	Select a project and select <b>Project &gt; Remove Development Path</b> or <b>History &gt; Remove Development Path</b> . In the <b>Remove Development Path</b> dialog box, select a development path to remove from the <b>Development Path</b> list.

**NOTE** You cannot remove a development path from a project if a checkpoint is in progress on that project. Additionally, you cannot remove a development path from build projects.

---

# Merging a Child Development Path

MKS Integrity provides you with a streamlined method of merging a development path into its parent development path. The parent development path may be mainline, or it may be a development path itself. The merge destination just needs to be the parent of the child being merged into it.

The following are key considerations when merging a development path:

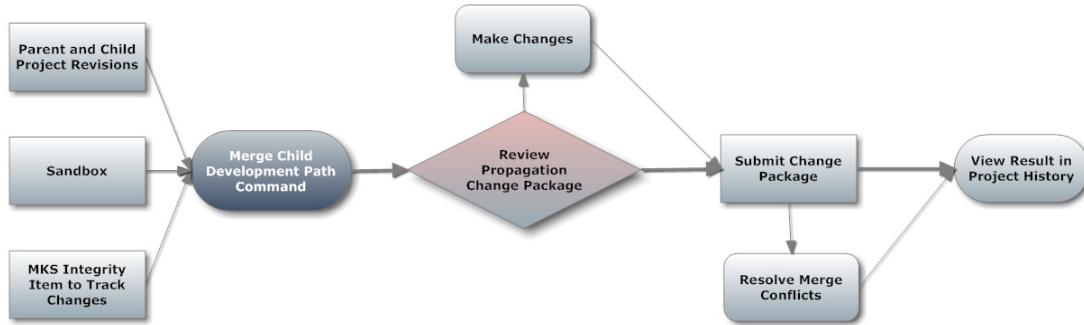
- This command requires a Sandbox that contains the project revision you want to merge into (parent project).
- Always start with the parent revision and merge into it
- If a project is selected (instead of a sandbox) you are prompted to select the sandbox.
- If you have previously performed a merge from one branch into another, when repeating the operation the command only picks up changes from change packages that were closed after the last merge (only new changes are merged).
- As a best practice, when making changes on multiple development paths, use a separate change package for each development path (ensure no change package contains entries from two or more development paths).
- For large merge operations, a significant number of smaller manual merge tasks may be required to resolve merge conflicts.
- As a best practice, the development paths being merged should be structurally the same. Any structural changes performed to a target of the development path merge operation after the source branch was created (such as configuring a subproject) can potentially cause problems during the merge.
- If the project structure contains the same project twice in the same level (for example, a subproject in the mainline and a subproject configured as a variant) then the merge is not recorded and the merge line does not display in the project history.
- When the propagation change package is submitted, MKS Integrity checkpoints the destination project. The destination project is locked during the duration of the checkpoint.
- If you are running a separate server for configuration management from the server used for workflows and documents, ensure the server clocks are synchronized.

## Merge Checklist

Ensure you have access to the following:

- Two project revisions: one, the parent (destination) project revision you intend to merge into; and two, the development path (child or source project revision) that is to be merged.
- Sandbox containing the parent development path.
- MKS Integrity item created to store the merge information and change package (may be optional depending on your MKS Integrity implementation).

# Merging a Development Path Overview



The following is a general overview of the merge development path process:

- 1 With the parent revision selected, start the merge development path operation, and then specify the child development path and an item to capture the changes (if workflows and documents are enabled).  
The command creates a propagation change package that contains the changes necessary to perform the development path merge.
- 2 Review the contents of propagation change package to ensure they are correct. Make changes to the change package entries if needed.
- 3 To complete the merge, submit the propagation change package. If necessary, resolve merge conflicts. When finished, view the result in the project history.

## Merging a Development Path Procedure

The procedure for merging a development path is divided into two tasks: the pre-merge task and the merge task.

### Pre-Merge Task

The following are the steps to perform to set up the development path merge operation before the merge actually takes place.

- 1 In the Sandboxes view, locate and select the project revision you want to merge a development path into, and then select the **Merge Child Development Path** action (command). The **Merge Child Development Path** dialog box displays.

**TIP** Whenever a sandbox is not in context, selecting Merge Development Path displays a Sandbox selection dialog box.

---

**2** Specify the following:

- Select the source development path you intend to merge into the parent development path. Only children of the previously selected project display in the list.
- Specify an MKS Integrity item to store merge information (and the propagation change package created by the merge operation). This list only displays if your implementation of MKS Integrity has the workflows and documents integration is enabled. Note that the list item <none> will not display if items are mandatory.

When you are finished, click **OK**.

The command creates a propagation change package that contains the changes necessary to perform the development path merge. The command does so by determining the list of change packages to propagate. The command finds all of the closed change packages that have at least one entry on that development path. If a change package has entries that are not on the project tree, then those entries are ignored.

If a change package has entries that are not on the development path but are still on the same project, then they are used (picked up) by the command. MKS recommends you do not use change packages with members across development paths.

If you previously performed a development path merge from one branch to another, when repeating the operation, the command filters out the list of change packages leaving only those change packages that were closed after the last merge completed.

When the command completes, the propagation change package is automatically created on the source branch.

If there is a merge conflict, a dialog box displays. You have the choice to merge automatically, or manually merge (consult the *MKS Integrity User Guide*). Clicking **OK** displays the propagation change package contents.

**3** View the propagation change package and review the changes to ensure they are as desired.

---

**IMPORTANT** At this point, no actual merging or changes have been performed to the projects.

---

You can also view the item for the change package, that contains the description of the propagation change package and operation's report summary.

**Merge Task**

To merge the development path, submit the propagation change package created in the pre-merge task. You may receive prompts for operations contained in the change package that require input.

Once the merge is completed, MKS Integrity checkpoints the destination (parent) project. You can verify the results of the merge by viewing the project history (**View Project History** action). The merge operation is represented by a merge line.

## When Should I Not Merge Development Paths?

The following is a list of conditions when you should not use the **Merge Child Development Path** (`si mergechilddevpath`) command:

- **Backfill**

Avoid operations requiring a backfill because it makes the operation difficult to yield the desired results.

---

- **Backward Propagations**

The command should only be used for forward propagations. For example, if member revision is 1.2, and you intend to resync to 1.4, the command responds correctly (as long as you are propagating forward on that development path). However, if you want to update member revision back to an earlier revision (for example, 1.1) the command will do nothing because it determines the change has already been made.

- **Large Propagations**

Keep the number of propagations small so that when they fail they are easier to resolve. Perform your development path merge operations as frequently as possible. For example, do not perform two years worth of operations because the accumulation of complexity makes it difficult for the operation to yield the desired results. You will be more successful performing the operation in small chunks.

- **Refactoring Operations**

Avoid refactoring operations such as move, rename, move subprojects. For example, if the configuration of a child development path is very different from the parent in terms of structure, then it will be more difficult for the merge operation to yield the desired results.

- **Unclosed Change Packages**

Ensure all change packages for the source development path have been closed after the last development path merge operation. The operation only uses changes from closed change packages. Similarly, ensure all change packages that are under review have been reviewed.

- **Non-transactional Change Packages**

Use transactional change packages. If possible, to avoid problems, make transactional change packages mandatory on your system. Ensure all changes needed for the propagation are in change packages, because if some required operations are outside of a change package and some are in a change package, the merge will fail (because it is unable complete some needed operations).

- **Re-used Development Path Names**

Do not re-use development path names. For example, when creating a new development path, do not use the same name of one that was previously deleted.

# Locating Where a Configuration Management Object Is Used

**CLI EQUIVALENT** `si locate`

As configuration management projects grow in complexity, it can be difficult to determine the relationships that MKS Integrity objects – projects, subprojects, and members – have with one another. Using the **Locate** command, you can locate where a specific MKS Integrity object is used in your configuration management repository. The search results appear in a **Locate** view, that allows you to refine and sort the results, view each located object in greater detail, and perform basic operations on them, such as viewing members.

When working with multiple development paths and shared subprojects or shared member archives, it becomes important to understand what areas of a project might be affected by parallel development. For example, you might need to:

- determine what active projects a selected member, subproject, or registered top-level project is shared with. Active projects are projects that are currently under development.
- determine what development paths a selected member or subproject belongs to
- determine if a selected member revision is part of a checkpoint
- confirm all existing checkpoints for a selected revision before you delete it
- display the development path that a corresponding member revision belongs to
- display which master projects a selected subproject belongs to

Operation	Procedure
To locate where a project is used	Select a project, then select <b>Project &gt; Locate</b>
To locate where a subproject is used	Select a project, then select <b>Project &gt; Subproject &gt; Locate</b>
To locate where a member is used	Select a member, then select <b>Member &gt; Locate</b>
To locate where a revision is used	Select a revision, then select <b>Member &gt; Revision &gt; Locate</b>
To locate where a subproject is used by searching by subproject name	Select a project, then select <b>Project &gt; Subproject &gt; Locate By Name</b>
To locate where a member is used by searching by member name	Select a member, then select <b>Member &gt; Locate By Name</b>

If you are searching for a member or subproject by name, You can specify the maximum number of results to display. The number of results displayed is also controlled by the server limit, which is set at 5000 by default. Administrators can change the default limit. If the number of objects that match the search criteria exceeds the set limit, MKS Integrity displays the following message at the bottom of the **Locate** view: Showing <set limit> of <total number of objects>.

## Key Considerations

- The **Locate** command can only be used in the database repository. Running the command from the file system repository returns an error message.
- You require the `OpenProject` permission for the project you are running the **Locate** command from. When the **Locate** view displays, MKS Integrity informs you if there are additional projects that you do not have permission to view.

- 
- Archive locations do not display in the **Locate** view.
  - Objects in development paths that have been removed do not display in the **Locate** view.

## Locate Options

Options	Description
<b>Depth for lookup</b>	<b>Depth for lookup</b> specifies the depth of the search. Select one of the following options: <ul style="list-style-type: none"> <li>■ <b>Current items</b> searches current project configurations (normal and variant projects).</li> <li>■ <b>Current and Checkpointed</b> searches current project configurations and checkpoints.</li> <li>■ <b>All items</b> searches current project configurations, checkpoints, and in between checkpoints.</li> </ul>
<b>Project Scope</b>	<b>Project Scope</b> specifies which projects to search. Select one of the following options: <ul style="list-style-type: none"> <li>■ <b>This Project</b> searches the current project and its subprojects.</li> <li>■ <b>Other Projects</b> searches all projects and subprojects, except the current project.</li> <li>■ <b>All Projects</b> searches all projects and subprojects.</li> </ul>
<b>Devpath Scope</b>	<b>Devpath Scope</b> specifies which development paths to search. Select one of the following options: <ul style="list-style-type: none"> <li>■ <b>This Devpath</b> searches the current development path.</li> <li>■ <b>Other Devpaths</b> searches all development paths, except the current development path.</li> <li>■ <b>All Devpaths</b> searches all development paths.</li> </ul>
<b>Default Mode</b>	<b>Default Mode</b> specifies the default mode that you want to display the search results in. Select one of the following options: <ul style="list-style-type: none"> <li>■ <b>Distinct Mode</b> displays only the projects and development paths containing the object you are searching. From the <b>Default Distinct</b> list, choose the information you want to display in <b>Distinct Mode</b>. If a subproject is shared by two or more projects, only the original project is displayed. Any projects where the subproject was added as shared are not displayed.</li> <li>■ <b>List Mode</b> displays detailed information about all occurrences of the object in the projects and development paths containing the object.</li> </ul>
<b>Default Distinct</b>	<b>Default Distinct</b> specifies the information to display in <b>Distinct Mode</b> . Select one of the following options: <ul style="list-style-type: none"> <li>■ <b>Projects</b> lists only project names.</li> <li>■ <b>Development paths</b> lists only development path names.</li> <li>■ <b>Registered Projects</b> lists only registered top-level project names.</li> </ul>
<b>Limit Search to Active Paths</b>	<b>Limit Search to Active Projects</b> searches projects that are accessible from head revision and are registered projects.

## Locate View

When you use the **Locate** command to locate where an MKS Integrity configuration management object is used, a **Locate** view displays the information.

The **Locate** view displays the following information:

Tab	Description
<b>Projects</b>	Distinct set of project names located by the search.
<b>Development Paths</b>	Distinct set of development path names located by the search.

Tab	Description
<b>Registered Projects</b>	Distinct set of top-level project names located by the search.
<b>All Occurrences</b>	<p><b>Checkpoints</b> displays the checkpoints containing the object.</p> <p><b>Name</b> displays the name of the object.</p> <p><b>Registered Projects</b> displays the top-level projects containing the object.</p> <p><b>Dates</b> displays the date ranges in which the object exists.</p> <p><b>Project</b> displays the projects containing the object.</p> <p><b>Development Path</b> displays the development paths containing the object.</p> <p><b>Revisions</b> displays the member's revisions.</p>

If other projects exist that you do not have permission to view, MKS Integrity displays the following message at the bottom of the view: Other records exist that you are not allowed to see.

If you are searching by name, you can specify the maximum number of results to display. If the number of objects that match the search criteria exceeds the set limit, MKS Integrity displays the following message at the bottom of the view: Showing <set limit> of <total number of objects>. The number of results displayed is also controlled by the server limit, which is set at 5000 by default. Administrators can change the default limit.

To change the depth and scope of the search, select **View > Change Settings**. The **Locate** dialog box displays, allowing you to refine your search.

---

**NOTE** It can take a long time to display registered projects on the **Registered Projects** tab. Information on the **All Occurrences** tab can also take a long time to display if the tab contains the **Registered Projects** column.

---

You can also perform some commands on objects selected in the **Locate** view. For example, you can view a selected member's history or delete a selected revision.

# Promoting and Demoting Configuration Management Projects

Administrators can define promotion states that control how development objects move through a structured development cycle. For example, a project could go through states such as: development, testing, production, and release. You promote a project to move it to the next state in the cycle, or demote it to return it to the previous state.

**NOTE** Promoting and demoting projects is for historical purposes only. To control project workflow more effectively, MKS recommends using MKS Integrity workflows and documents.

## **Promoting a Project**

**CLI EQUIVALENT** `si promoteproject`

Typically, you promote a project (or set its state) when you checkpoint it, but you can do so at any time. To promote a member, your administrator must have defined promotion states.

**NOTE** When you promote a project, only the project (.pj) file is affected. Individual members of the project are not changed.

Interface	Procedure
GUI	Select a checkpoint from the <b>Project History view</b> and select <b>Project &gt; Properties &gt; Promote</b> . In the <b>Promote Project</b> dialog box, select a new state from the <b>Promote to State</b> list.
Web	Select a checkpoint from the <b>Project History view</b> and click <b>History &gt; Promote</b> . In the <b>Promote Project</b> dialog box, select a new state from the <b>Promote to State</b> list.

**TIP** In the configuration management Web interface, you can also promote a project from the **Project** view by selecting **Project > Promote**.

## **Demoting a Project**

**CLI EQUIVALENT** `si demoteproject`

Typically, you demote a project (or set its state) when you checkpoint it, but you can do so at any time. To demote a member, your administrator must have defined promotion states.

---

When you demote a project, only the project file (.pj) is affected. Individual members of the project are not changed.

---

**NOTE** Demoting projects is for historical purposes only. To more effectively control project workflow, MKS recommends using MKS Integrity workflows and documents.

---

Interface	Procedure
To demote a project in the GUI	Select a checkpoint from the <b>Project History view</b> and select <b>Project &gt; Properties &gt; Demote</b> . In the <b>Demote Project</b> dialog box, select a new state from the <b>Demote to State</b> list.
To demote a project in the Web interface	Select a checkpoint from the <b>Project History view</b> and select <b>History &gt; Demote</b> . In the <b>Demote Project</b> dialog box, select a new state from the <b>Demote to State</b> list.

---

**TIP** In the configuration management Web interface, you can also demote a project from the **Project** view by selecting **Project > Demote**.

---

## PART 10

# Managing Your Personal Workspace in a Sandbox

---

# Sandboxes Overview

A Sandbox is a private workspace that resides on the client machine and mirrors the content of a project on the server. Although it looks and acts like the project it mirrors, a Sandbox is actually a collection of pointers to its real-life counterparts in the master project. Sandboxes allow you to work locally in your own workspace, without interfering with the work of others.

Different types of Sandboxes are available for different types of development.

- *Normal* Sandboxes are useful for the sequential development of a project over the long or short term.
- *Variant* Sandboxes are useful for branching off the main development path.
- *Build* Sandboxes are useful for testing a specific revision of the project.

## **Using Variant Sandboxes**

A *variant Sandbox* is based on a specific development path of a project. When you create a variant Sandbox, you choose the development path to use. In the variant Sandbox, you can see the current state of the project along that development path and the changes made by other developers using it.

When a variant Sandbox is created for the first time, it is also created for all subprojects, reserving the assigned name as a unique identifier and ensuring no two paths can share the same name.

Conflicts can occur when developers working on different paths need to work on the same revision of a file. For example, one developer could be working in a regular Sandbox that includes `utility.dll`, version 1.4 and another developer could be working in a variant Sandbox that contains `utility.dll`, version 1.3. Both versions are stored in the same member history.

To prevent potential conflicts, the first time you check in a member from a variant Sandbox, you are prompted to branch the member history. Branching the member history gives each development path its own copy of the revision.

## **Using Build Sandboxes**

After major milestones, such as product releases, you might want to recreate a static version of an entire project as it existed at some point in the past. You create a build Sandbox to build or test the project, but not to begin further work along a new development path. Build Sandboxes could be used for quality assurance or production to distribute files in a fixed configuration.

A *build Sandbox* is a Sandbox associated with a particular project checkpoint and has no development path (since it is static and not meant for further development). No further development can be carried out in a build Sandbox.

For example, if a build manager needed to burn a CD of a special build that did not include a certain feature, he could use an earlier checkpoint to create a build Sandbox on the CD burning system.

Within a build Sandbox, you can:

- change labels and states
- resynchronize your Sandbox
- compare a member revision in the build Sandbox to another revision
- merge a member revision in the build Sandbox with another revision (of course, you cannot check a merged file back into the build Sandbox)

- 
- check for differences between checkpoints, such as changes to a project since it was last checkpointed

When you create a build Sandbox, you choose the project checkpoint to base the build Sandbox on.

However, with a build Sandbox, you cannot:

- check out, lock, or check in members
- add or remove members
- set the development path
- freeze or thaw members
- checkpoint the master project
- modify project or member attributes
- revert members
- set the member revision

Each of these represents further development, which requires a normal or variant Sandbox.

---

# My Sandboxes View: Managing Sandboxes

---

**CLI EQUIVALENT** `si sandboxes`

---

To view your sandboxes, select **Sandbox > View My Sandboxes**.

From the **My Sandboxes** view you can open, create, import, and drop Sandboxes.

The **My Sandboxes** view contains three tabs that display the following different types of Sandboxes:

- **Regular** is a Sandbox used for normal or main project development.
- **Variant** is a Sandbox used for branching off the main development path of a project.
- **Build** is a Sandbox based upon a specific revision of a project, typically used as a reference for building or testing the project. It is not suitable for further development due to limited project modification abilities.

---

**NOTE** Sandboxes based on configured subprojects do not appear with a configuration icon in the **My Sandboxes** view.

---

# Viewing Sandbox Details

**CLI EQUIVALENT** `si sandboxinfo`

To display detailed information for a Sandbox, select **Sandbox > Views > View Information**. The **Sandbox Information** view displays.

The **General** tab displays the following general Sandbox information:

- **Sandbox Name** is the path and name of the Sandbox.
- Pending information displays if the project you selected is a subproject that is pending, has a pending subproject operation, or is located within a pending subproject.
- **Project Name** is the path and name of the master project.
- **Server** is the MKS Integrity Server name and port number where the master project resides.
- **Configuration Path** displays the configuration path of the project. The following keywords may appear in the path:
  - # specifies a project or subproject in a well-formed project tree.
  - #p specifies a project that does not end with `project.pj`.
  - #n specifies a normal project.
  - #s specifies a subproject in a poorly formed project tree.
  - #d specifies the development path name.
  - #b specifies the number, label, or symbolic of the project checkpoint.
- **Type** display information describing the use of the project in Deploy. You only see this field if you are licensed to use Deploy. For more information, see the *MKS Deploy 2009 Administration Guide*.
- **Members** is the number of members in the Sandbox.
- **Subsandboxes** is the number of sub Sandboxes in the Sandbox.
- **Sparse** setting determines if the selected Sandbox is a sparse Sandbox.
- **Shared** setting determines if the selected Sandbox is a shared Sandbox.
- **Scope** displays the current Sandbox scope definition (if specified) and allows you to change the scope. Sandbox scope defines what project members are included in the Sandbox, transferring specific members from the MKS Integrity Server to the Sandbox directory when the Sandbox is created and controlling what members display in the Sandbox view.

To change the Sandbox scope definition, ensure that you are viewing the Sandbox information for the top-level Sandbox and click **Change Scope**.

Changes to the scope definition are automatically reflected in the Sandbox view. Members with working files in the Sandbox that no longer match the scope definition display deltas, but remain in the Sandbox. Selecting one of the members indicates that a working file exists and the member does not match the Sandbox scope. Performing a **Resynchronize** or **Revert** operation removes the members from the Sandbox. For more information, see “Specifying the Sandbox Scope” on page 291.

---

**TIP** Using the `si createsandbox` command, you can create and edit more complex Sandbox scope definitions using a combination of logical AND or OR operators; however, these definitions may not always be editable from the GUI. If you attempt to edit a complex scope definition from the GUI, MKS Integrity truncates the definition to what the GUI is capable of displaying. If you attempt to edit a complex scope definition using the `si configuresandbox -g/gui` command, MKS Integrity displays a warning message that choosing to edit the scope definition removes all options the GUI is not capable of displaying.

- **Line Terminator** setting determines the type of ASCII character MKS Integrity recognizes as the end of a line of text: native (or automatic, the default setting), `\f` (or line feed, primarily for UNIX applications), `\r`, or `\r\n`.
- **Project Description** is a description of the master project.
- **Last Checkpoint** displays the following information about the last checkpoint of the master project: the revision number of last checkpoint, the date and time of last checkpoint, and a brief description of the last checkpoint. You cannot change an existing checkpoint description from this dialog box, but you can append additional comments to it. To do so, enter any supplemental information in the **Checkpoint Description** field below the present information.

The **Project Attributes** tab displays the attributes that apply across the entire project. Attributes can conform to either of the following formats:

- *variable*
- *variable=value*

The **Sandbox Attributes** tab displays the attributes that apply to your workspace. You can modify Sandbox attributes. Attributes conform to either of the following formats:

- *variable*
- *variable=value*

The **Associated Items** tab displays the ID, type and summary of any MKS Integrity items associated with the project. Only item types that you have permission to view are displayed. You can right-click an associated item to view or edit it.

# Creating a Sandbox

---

**CLI EQUIVALENT** `si createsandbox`

---

Once a Sandbox is created, you can add files or members to that Sandbox. The project is then updated to reflect the addition of new project members.

To create a Sandbox in the GUI, select **Sandbox > Create** and follow the instructions in the **Create Sandbox Wizard**.

---

**NOTE** While it is possible to create more than one Sandbox in a single directory, it is not recommended.

---

## **Specifying the Project to Base the Sandbox On**

You can type in or select the project to base the Sandbox on.

If you are creating a variant or build Sandbox in the GUI, initially only enter the path and name of the root project. You specify the subproject later in the procedure. When specifying the subproject, there are rules that control what project configuration you can jump to. If your selection breaks any of the rules, you cannot create the Sandbox.

## **Selecting a Sandbox Type**

When creating a Sandbox you can create the following types:

- **Normal** creates a Sandbox based on the working project on the mainline.
- **Variant** creates a Sandbox based on a project on a specific development path.

---

**NOTE** The **Variant** option is unavailable if there are no available development paths.

---

- **Build** creates a Sandbox based on a specific checkpoint of the master project. You can specify the checkpoint through its checkpoint number or label.

---

**NOTE** If the project that the Sandbox is based on is specified in the **Project Name** field using a keyword-based string with a jump to a variant or build project, you cannot select the type of Sandbox to create. When you click **Finish**, a variant or build Sandbox is created for you based on the configuration of the specified project.

---

## **Specifying the Sandbox Scope**

For large projects with mixed content, such as source code, simulation files, calibration files, and documentation, creating complete Sandboxes from these projects can impact MKS Integrity Client performance and consume large amounts of network bandwidth. While you may need all components in the projects (subprojects), you may only need to work with certain files or file types.

Specifying a Sandbox scope allows you to define what project members are included in a Sandbox, transferring specific members from the MKS Integrity Server to the Sandbox directory when the Sandbox is created and controlling what members display in the Sandbox view. Specifying Sandbox scope can greatly improve the performance of the Create Sandbox operation, reduce network traffic, and make it easier to locate the relevant members that you need to work with.

In the **Sandbox Wizard**, under **Options > Sandbox Scope**, click **Change Scope**.

Click a checkbox to include () or invert () one or more of the following options:

Option	Description
<b>All Members</b>	All project members. This is the default setting.
<b>Members with Attribute</b>	Project members with an attribute or attribute set to a value, for example, Beta or OS=Windows. This option is case-sensitive.
<b>Members with Path</b>	<p>Project members that reside in a directory, relative to the top-level Sandbox, for example, <code>watch/lib/*</code>. The specified path does not differentiate between subdirectories and subproject names. This means that you cannot specify individual co-located subprojects.</p> <p>For example, if you create a scoped Sandbox from the following top-level project:  <code>/p1/project.pj</code>  with the following subprojects and members:  <code>/p1/sub1/project.pj</code>  <code>/p1/sub1/aa.txt</code>  <code>/p1/sub1/bb.txt</code>  <code>/p1/sub1/dd.txt</code>  <code>/p1/sub2/project.pj</code>  <code>/p1/sub2/sub1/cc.txt</code>  specifying <code>sub1</code> matches <code>p1/sub1/aa.txt</code> and <code>p1/sub1/dd.txt</code>  or specifying <code>*sub1</code> matches <code>/p1/sub1/aa.txt</code>, <code>/p1/sub1/bb.txt</code>, <code>/p1/sub2/sub1/cc.txt</code>, and <code>/p1/sub1/dd.txt</code>.</p> <p><b>Note:</b> If the client OS is a case-sensitive file system and the database repository on the server is case-sensitive, this option is case-sensitive. Otherwise, this option is case-insensitive.</p>
<b>Members with Name</b>	<p>Project members with a name or file extension, for example, <code>Readme.txt</code> or <code>*.java</code>. A name is only valid for a file name, not a leading directory prefix.</p> <p><b>Note:</b> If the client OS is a case-sensitive file system and the database repository on the server is case-sensitive, this option is case-sensitive. Otherwise, this option is case-insensitive.</p>
<b>Member with Label at Member Revision</b>	Project members with a label at member revision, for example, <code>TEST</code> . This option is case-sensitive and accepts wildcards (*) and (?).
<b>Members with Label at Any Revision</b>	Project members with a label at any revision, for example, <code>PROD</code> . This option is case-sensitive and accepts wildcards (*) and (?).
<b>Members with Archive Type</b>	Project members that are a <code>Binary</code> or <code>Text</code> archive type.
<b>Combine Selections Using</b> <ul style="list-style-type: none"> <li>■ <b>Logical AND</b></li> <li>■ <b>Logical OR</b></li> </ul>	<p>Combines multiple Sandbox scope options using a logical AND or OR operator. For example, to include project members with member attribute <code>Beta</code> AND name <code>*.java</code>, set <b>With Attribute</b> to <code>Beta</code>, <b>Members with Name</b> to <code>*.java</code>, and enable <b>Logical AND</b>.</p> <p><b>Tip:</b> Using the <code>si createsandbox</code> command, you can create and edit more complex Sandbox scope definitions using a combination of logical AND or OR operators; however, these definitions may not always be editable from the GUI. If you attempt to edit a complex scope definition from the GUI, MKS Integrity truncates the definition to what the GUI is capable of displaying. If you attempt to edit a complex scope definition using the <code>si configuresandbox -g/gui</code> command, MKS Integrity displays a warning message that choosing to edit the scope definition removes the options the GUI is not capable of displaying.</p>

After the Sandbox is created, the title bar in the Sandbox view displays **Scoped Sandbox path and project**.

You can view and change the scope definition from the **Sandbox Information** dialog box or **Configure Sandbox** command. Changes to the scope definition are automatically reflected in the Sandbox view. Members with working files in the Sandbox that no longer match the scope definition display deltas, but remain in the Sandbox. Selecting one of the members indicates that a working file exists and the member does not match the Sandbox scope. Performing a **Resynchronize** or **Revert** operation removes the members from the Sandbox.

---

Note the following:

- The scope of a Sandbox is saved with the Sandbox and persists when the Sandbox or MKS Integrity Client are closed and restarted. It is possible to have two Sandboxes of the same project with each one having a different Sandbox scope.
- For future Create Sandbox operations, you can set the Sandbox Scope in the preferences for the **Create Sandbox** command.
- New project members are affected by the Sandbox scope. For example, if your Sandbox scope includes project members with attribute name and value `OS=Windows`, and you add a project member with that attribute name and value in a non-scoped Sandbox, the member displays in the scoped Sandbox as a new member. Resynchronizing the member adds the working file to the Sandbox. If you add a project member without an attribute name and value in non-scoped Sandbox, the member does not display in the scoped Sandbox.
- Defining the Sandbox scope is different from filtering members using the **Filter** list at the top of a Sandbox view. The Sandbox scope determines what members initially display in the Sandbox. The **Filter** list allows you to further filter those Sandbox members. For example, if your Sandbox scope includes members with the attribute name and value `OS=Windows`, and there is a new project member locked by another user that does not contain that attribute name and value, the `Locked Members` filter in the **Filter** list does not display the locked member because it does not match the Sandbox scope, even though the member matches the `Locked Members` filter.
- The scope and sparse options are independent of one another, for example, you can create a scoped sparse Sandbox.
- In a shared scoped Sandbox, other users see the Sandbox scope, but cannot edit the scope.
- You can manage out of scope members by using the Sandbox **Filter** list and the **View > Select** command. To display only members that have working files in the Sandbox that do not match the current Sandbox scope definition, select `Out of Scope Members` from the Sandbox view **Filter** list. To select only members that have working files in the Sandbox that do not match the current Sandbox scope definition, select **View > Select** and enable the **Out of Scope Members** option.

# Configuring a Sandbox

---

**CLI EQUIVALENT** `si configuresandbox`

---

If you want to change the configuration of a Sandbox, open the **Sandbox Information** view for a top-level Sandbox created by you (**Sandbox > Views > View Information**), and check the appropriate options on the **General** panel:

- **Sparse** if you do not want the Sandbox to retain working files when a member is checked in. A Sandbox is sparse if it has no working files. A sparse Sandbox does not retain working files when a member is checked in, and continues to function this way throughout its use; however, once created, Sandbox directories and sub Sandboxes remain in the Sandbox.
- **Shared** if you want multiple users to have access to the working files in the Sandbox.
- **Scope** displays the current Sandbox scope definition (if specified) and allows you to change the scope. Sandbox scope defines what project members are included in the Sandbox, transferring specific members from the MKS Integrity Server to the Sandbox directory when the Sandbox is created and controlling what members display in the Sandbox view.

To change the Sandbox scope definition, ensure that you are viewing the Sandbox information for the top-level Sandbox and click **Change Scope**.

Changes to the scope definition are automatically reflected in the Sandbox view. Members with working files in the Sandbox that no longer match the scope definition display deltas, but remain in the Sandbox. Selecting one of the members indicates that a working file exists and the member does not match the Sandbox scope. Performing a **Resynchronize** or **Revert** operation removes the members from the Sandbox. For more information, see “Specifying the Sandbox Scope” on page 291.

---

**TIP** Using the `si createsandbox` and `si configuresandbox` commands, you can create and edit more complex Sandbox scope definitions using a combination of logical AND or OR operators; however, these definitions may not always be editable from the GUI. If you attempt to edit a complex scope definition from the GUI, MKS Integrity truncates the definition to what the GUI is capable of displaying. If you attempt to edit a complex scope definition using the `si configuresandbox -g/gui` command, MKS Integrity displays a warning message that choosing to edit the scope definition removes the options the GUI is not capable of displaying.

---

# Opening a Sandbox

Opening a Sandbox in the GUI allows you to view and work with the Sandbox and its members. The **Sandbox** view displays the members and sub Sandboxes of a single Sandbox.

**NOTE** To see a list of recently used Sandboxes, select **Sandbox > Recent**. To open a Sandbox, select one from the list.

**TIP** You can view which Sandbox locked a member from the **Member History** view, **Locks** view, **Project** view, and **Sandbox** view by setting the view preferences to display the information.

To open a Sandbox in the GUI, select **Sandbox > Open Sandbox** and select the type of Sandbox to open:

- **Normal** opens a Sandbox based on the working project on the mainline.
- **Variant** opens a Sandbox based on a project on a specific development path.

**NOTE** The **Variant** option is unavailable if there are no available development paths.

- **Build** opens a Sandbox based on a specific checkpoint of the master project. You can specify the checkpoint through its checkpoint number or label.

## Sandbox View

**CLI EQUIVALENT** `si viewsandbox`

When you open a Sandbox, MKS Integrity displays its contents in a **Sandbox** view.

*Sandboxes* can reside on the client machines and allow you to work locally in your own workspace without interfering with the work of others. You can think of the Sandbox as a local pointer to the project residing on the MKS Integrity Server. A Sandbox is a mirror image of an MKS Integrity SCM project. Although it looks and acts like the project it mirrors, it is actually a collection of pointers to its real-life counterparts in the master project.

The **Sandbox** view displays project members, and sub Sandboxes in a tree structure. You can expand and collapse a project, Sandbox, and sub Sandbox tree using one of the following options:

- Double-click the Sandbox or sub Sandbox.
- Click the plus or minus icons in the directory list.
- Right-click the Sandbox or sub Sandbox and select **Expand All**.
- Select **View > Expand All** or **View > Collapse All**.

Select a member by clicking it or by moving the selector bar to it with the cursor control keys. Use the up and down arrows to move through the project or Sandbox hierarchy. Use the right arrow to open a project or Sandbox, revealing its members. Use the left arrow to close a project or Sandbox.

A box below the list of members displays all delta information: working file deltas, new revision (member) deltas, and revision sync deltas. For example, if the working file has been updated but not checked in, a message tells you that the working file has been changed. The box also displays details on the working file and member revision numbers that are available in the project. When there are size differences between the working file and member revision, the member information pane also displays this difference (in kilobytes).

---

You can view and work with a group of project members based on specific member properties. You can filter the members displayed using the **Filter** list located on the toolbar or select a group of members using **View > Select**.

If you have configured the **Sandbox** view to be dynamic, it is automatically updated to reflect changes in the **My Sandboxes** view. The information in Sandbox views displays in columns with headings and icons. Depending on your view preferences, the columns displayed may be different than the default preferences described here.

The following table outlines the default columns and what they display:

Column	Description
<b>Name</b>	Displays the name of the project, Sandbox, or member.
<b>Working Rev.</b>	Displays the checked out revision number corresponding to the member's working file, for example, 1.2. If it is a pending revision, pending displays in parenthesis, for example, 1.2 (pending). If the working file is locked and out of sync, a gray non-exclusive lock icon displays.
<b>Member Rev.</b>	Displays the member's revision number, or in the case of a subproject, the subproject revision number (if any), for example, 1.3.
	Indicates that your working file has changed. A delta symbol with a white document icon means your working file has changed. A delta symbol with a blue striped document icon indicates that the working revision does not match the member revision. A description of the changes is shown at the bottom of the window.
	A snowflake icon means the member is frozen. When the member is thawed, the snowflake icon disappears.
	A blue striped document with a hand icon means the member shares another member's archive. This column is valid only if you are using the database repository.
<b>Locked</b>	Users who have a lock on the member. The padlock icon indicates the type of lock and any potential conflicts with other lockers. If there are multiple lockers, your lock is listed first, followed by the locker who has an exclusive lock (if any), followed by any lockers with non-exclusive locks. If there are multiple lockers, click on the field to display a drop-down list of lockers. <b>Note:</b> Only locks on the head revision display. If there are multiple lockers, and another locker checks in before you, your lock no longer displays because it is not on the head revision. For more information on your locks policy, contact your administrator.
<b>Labels</b>	Displays any labels assigned to the member revision, for example, Draft1.
<b>State</b>	Displays the state of the member revision, for example, Beta.
<b>Working CPID (Sandbox only)</b>	Displays the change package associated with a deferred or lock operation performed by the current user from the current Sandbox.
<b>Member CPID</b>	Displays the change package associated with the operation that set the member's revision. If you do not specify a change package for the operation, this column is not updated.
<b>Member Archive</b>	Displays the name of the archive the member refers to.
<b>Working Archive</b>	Displays the name of the archive the working file is derived from. <b>Note:</b> In the case where the working file was derived from an archive that was based on a server-side system setup (i.e. based on some default RCSPPath or WorkToArch setting, as opposed to an explicit archive= reference), the <b>Working Archive</b> field displays the value "(default:<server directory>)". The actual name of the archive from which the working file was derived can then be determined from the value for the <b>Member Archive</b> field.

---

# Importing a Sandbox

---

**CLI EQUIVALENT** `si importsandbox`

---

Use the Import Sandbox command when:

- you need to use a dropped Sandbox again
- you are sharing a Sandbox with another user
- your client registry was lost (for example, during a client machine upgrade)
- you need to change the project or server used by your Sandbox

---

**NOTE** The master project of the Sandbox must also be registered with the MKS Integrity Server before the Sandbox is imported.

---

To import a Sandbox in the GUI, select **Sandbox > Import** and browse to the Sandbox file on your local drive.

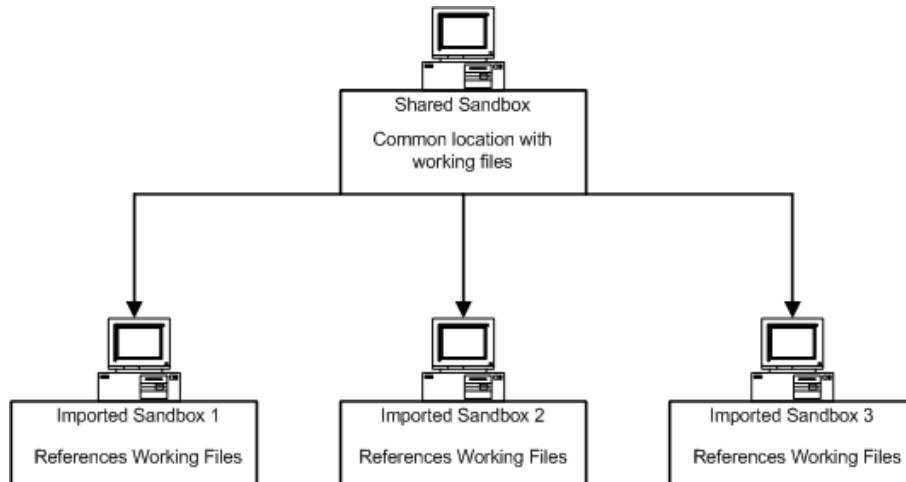
# Sharing Sandboxes

MKS Integrity provides a way to create a common build location using shared Sandboxes. Shared Sandboxes provide developers and buildmasters with a window into a single shared work location.

Sandbox sharing is intended when:

- There is a common build location that contains files that developers need to access to build their source before checking it into the build
- Users share the Sandbox by importing it through the network or a mapped drive. The imported Sandbox is by reference, so no working files are stored on the users' machines.
- Users who imported the shared Sandbox now have access to the working files in the shared Sandbox and use those files to create test builds on their machines without requiring those files to be stored locally.

**NOTE** MKS recommends that all clients connecting to the shared Sandbox be the same version of MKS Integrity.



*Sandbox shared with multiple users*

**CAUTION** MKS does not support using shared Sandboxes for collaborative development with multiple developers. Shared Sandboxes may become unstable in this scenario.

Users sharing Sandboxes:

- must select **View > Refresh** for the project to update their **Sandbox** view to see the true state of the Sandbox (top level project updates recursively)
- should all be in same time zone
- must have read and write permission for the shared location where the shared Sandbox resides, as well as for the files in that location
- cannot submit deferred operations in the Sandbox made by other users using the **Submit Change Package** command (their deferred and lock entries do not appear in your **Change Package** view)
- only top level Sandboxes can be shared (**Shared**, **Sparse**, and **Line Terminator** options in the **Sandbox Information** view are disabled for sub Sandboxes)

---

### **Configuring a Sandbox as Shared**

For users to share a Sandbox, only the creator of the Sandbox can configure the Sandbox as shared. When a top-level Sandbox is configured to be shared, all sub Sandboxes are shared as well. Sub Sandboxes cannot be individually configured to be shared or not shared.

To share a Sandbox in the GUI, open the **Sandbox Information** view for a top-level Sandbox created by you (**Sandbox > Views > View Information**), and check the **Shared** option on the **General** panel.

---

**IMPORTANT** If clients on different platforms are sharing a Sandbox, you may need to modify the line terminator setting to ensure compatibility with file editors.

---

### **Importing a Shared Sandbox**

You access another user's shared Sandbox using the **Import** command, and browsing through the network to the user's shared Sandbox directory. When you share another user's Sandbox, no working files are stored on your local machine.

Only import another user's Sandbox through the network if that Sandbox has sharing enabled.

Only top-level Sandboxes can be imported as shared.

### **Removing Sharing From a Sandbox**

When there is no longer a need for a common location, the creator of the Sandbox can remove the sharing from the Sandbox. When Sandbox sharing is removed, users sharing that Sandbox no longer have access to it.

Only the creator of the shared Sandbox can remove sharing (or edit Sandbox information).

To remove sharing from a Sandbox in the GUI, open the **Sandbox Information** view and clear the **Shared** option on the **General** panel.

Users sharing the Sandbox each receive an error message stating the change in sharing configuration the next time they attempt to access the Sandbox or perform an operation on its contents.

---

# Dropping a Sandbox

---

**CLI EQUIVALENT** `si dropsandbox`

---

When a Sandbox has outlived its usefulness or just does not belong anymore, you can remove it at any time.

To *drop* a Sandbox means that the Sandbox is no longer registered with the MKS Integrity Client. Once a Sandbox is dropped, it cannot be accessed with MKS Integrity. However, if you do not delete the Sandbox, you can import it again if you want it to be accessible in MKS Integrity.

You can select what files you want to delete when you drop your Sandbox:

- Select `Nothing` to not delete any files from the Sandbox directory on your local drive.
- Select `Sandbox Members Only` to delete the Sandbox members on your local drive but leave any other files.
- Select `Entire Sandbox Directory` to delete the Sandbox directory on your local drive and all of its contents.

---

**CAUTION** If you select the `Entire Sandbox Directory` option, MKS Integrity deletes all files in the Sandbox directory and all files in subdirectories, if any exist. This includes files not created by MKS Integrity.

---

---

# Retargeting a Sandbox

---

**CLI EQUIVALENT** `si retargetsandbox`

---

If you need to change the project configuration that a Sandbox points to, you can *retarget* the Sandbox. For example, a developer can change a variant Sandbox to point to a different variant project, or a buildmaster can change a build Sandbox to point to a different build. You can also change the type of project that the Sandbox points to; for example, you can change a normal Sandbox to point to a variant project.

You do not use the Retarget Sandbox command to change the project or server for a Sandbox or the location of a Sandbox on the client. To perform these operations, you must drop the Sandbox and then import it.

To retarget a Sandbox, select a top-level Sandbox and click **Sandbox > Retarget**, then select the project configuration that you want to point the Sandbox to.

## ***Key Considerations***

- You cannot retarget a shared Sandbox.
- You can only retarget top level Sandboxes.
- You can only retarget to another type of the same project that the Sandbox is currently pointing to.

# Taking Sandbox Snapshots

---

**CLI EQUIVALENT** `si snapshot`

---

A *snapshot* captures the current state of a Sandbox, where each element in the Sandbox can be identified with a pre-existing entity in the repository on the MKS Integrity Server. The Sandbox snapshot creates a project checkpoint that you can create a build Sandbox or development path from. The revision number of a checkpoint created by a snapshot includes the revision number of the last checkpoint. For example, if the last checkpoint of the project has a revision number of 1.1, then the revision number of the checkpoint created by the snapshot is 1.1.1.1.

To take a Sandbox snapshot, select a Sandbox, and then select **Sandbox > Snapshot**.

When you take a Sandbox snapshot, you can specify a state or a label for the snapshot, and apply it to all Sandbox members. A label is a unique text string assigned by you to identify the project checkpoint created by the snapshot. Labels cannot contain colons (:), square brackets ([ ]), or leading spaces. Additionally, they cannot have the same format as a valid checkpoint number.

The Sandbox snapshot displays as a branched project checkpoint in the **Project History** view.

---

**NOTE** If working files are missing from the Sandbox, a warning displays listing the missing working files that will not appear in the snapshot. If you want to include those working files in the snapshot, cancel the operation, provide the working files (by resynchronizing the corresponding members), and then perform the snapshot.

---

## Contents of a Sandbox Snapshot

The set of Sandbox elements captured in a snapshot includes the following:

- Sandbox members identified with an archive and working revisions the archive was created from
- former members that were dropped but are still present in your Sandbox
- sub Sandboxes identified by project name and type
- former sub Sandboxes that were dropped but are still present in your Sandbox

## Example

Steve has been assigned to create a build of the savings calculator. He uses a build Sandbox to perform the build. He discovers a readme file is missing from the checkpoint his build Sandbox is based on. Jen has added the readme file in a change package. Steve adds the readme file to his Sandbox using the change package and then takes a snapshot of his Sandbox to save the configuration.

## Key Considerations

- To be included in the snapshot, you cannot have working file changes in the Sandbox.
- If the working file revision differs from the member revision, it is the working file revision that is included in the snapshot.
- Members without working files are not included in the snapshot.
- Former members that still have working files in the Sandbox directory appear as members in the snapshot.
- Former subprojects that are still in the **Sandbox** view appear as subprojects in the snapshot.

- 
- MKS Integrity always uses the actual name of the member working file for the snapshot.
  - You cannot take a snapshot of a sparse Sandbox.
  - The **Snapshot Sandbox** command is performed on the entire Sandbox independently of the filter used to display the contents of a Sandbox.
  - You can compare differences between a project checkpoint created by a snapshot and another project checkpoint (including checkpoints created by a snapshot) in the project history, but you cannot compare differences with the Sandbox contents.
  - To specify an existing development path when taking a Sandbox snapshot, you must use the CLI. For more information, see the *MKS Integrity 2009 CLI Reference Guide for Configuration Management*.
  - Members of a Sandbox need to be associated with a corresponding archive on the MKS Integrity Server.
  - When recursing into sub Sandboxes, the snapshot represents exactly the same directory structure and files of your Sandbox. All subproject elements become the same type and shared subprojects of different types become shared subprojects of the same type.

When you take a snapshot of a Sandbox recursively that contains sub Sandboxes, the snapshot creates a checkpoint for the sub Sandboxes based on the last checkpoint of the master project (if one exists), not on the current subproject in your Sandbox. Member revisions are unaffected.

### **Using Sandbox Snapshots in a Development Environment**

The recommended scenario for when to take a Sandbox snapshot in a development environment is as follows:

- 1 You are in a situation where you are working in a regular Sandbox, but should be working in a variant Sandbox.
- 2 Instead of checking in your changes to the main development path, check in (or merge in) your changes on a branch.
- 3 Take a snapshot of the Sandbox.
- 4 Create a development path from the project checkpoint that corresponds to the snapshot.
- 5 Create a variant Sandbox from the development path you created, and then continue work on that development path.

---

**TIP** From the CLI, you can specify an existing development path at the time you take the snapshot. For more information, see the *MKS Integrity 2009 CLI Reference Guide for Configuration Management*.

---

### **Using Sandbox Snapshots in a Build Environment**

The recommended scenario for when to take a Sandbox snapshot in a build environment is as follows:

- 1 Checkpoint the project.
- 2 Create a build Sandbox for the build.
- 3 The build fails, but since development has continued, some of the required members are at later revisions than the last checkpoint.
- 4 Resynchronize the required revisions to fix the build (you can use Resync CP).
- 5 Take a snapshot of the Sandbox, and use the project checkpoint created by the snapshot to recreate the build in the future using a build Sandbox, instead of using the original project checkpoint.

PART 11

# Managing Source Files as Members

# Adding Members to a Project

---

**CLI EQUIVALENT** `si add`

---

Once you create a project, you can add files to it, making them members of that project. Once you add members to your project, you can check them out for editing, and then check them in to preserve your changes.

Project members may be one of following types of files: archived (files under MKS Integrity revision control) or subproject (another MKS Integrity SCM project). When adding members in the GUI, you have the option of adding members recursively to the project or creating subprojects.

---

**NOTE** To add server-side files to a project, use the Import Member command.

---

In the GUI, you can add members through the Sandbox. In the Web interface, because Sandboxes do not exist, you can add members through the project only.

Interface	Procedure
GUI	Select <b>Member &gt; Add</b> . Follow the instructions on the <b>Add Member Wizard</b> .
Web	Select <b>Project &gt; Add Member</b> and enter the name of the new member, including the file extension. For example, <code>features.gif</code> .

---

**TIP** To add selected non-member files to a project or subproject by using the drag-and-drop method, with a **Sandbox** view open drag the files from the **Non-Members** view to the Sandbox or sub Sandbox that corresponds to that project or subproject you want to add the project to. Subdirectories that do not correspond to subprojects are not maintained.

---

**NOTE** If you are using a file system repository, the maximum size for members is 2 GB; if you are using a database repository, the maximum size is 2 GB for Oracle or SQL Server, and 1 GB for DB2. To find out the type of repository used on your server, see your administrator.

---

## Key Considerations

- If you select multiple files to add in the GUI, they will be added to the project that they are closest to (top-level project or subproject).

---

**NOTE** Adding a large number of members or a directory with a large number of members to a Sandbox may cause the client to stall if the MKS Integrity Server does not have the recommended system configuration. Consult your administrator for more information.

---

- You can record the add member or create subproject operation in a change package. If you choose to create subprojects, the specified change package is used to record the Create Subproject operation.
- You can add a member to the project on a deferred basis. Deferring the addition of a member allows you to work with the member in your Sandbox without causing any changes in the project.

---

**NOTE** If there is a deferred add operation associated with a member, you cannot view the member history for that member because it has not yet been added to the actual project.

---

- If the member is being added to a deploy project, you can specify the deploy type for the member when entering the archive description in the **Create Archive** dialog box. The **Deploy Type** field only displays if you are licensed to use Deploy. For more information, see the *MKS Deploy 2009 Administration Guide*.
- MKS does not recommend adding symbolic link files to a deploy project. If you add a symbolic link file to a deploy project, it is deployed as a properties file containing a description of the link, rather than as the symbolic link file. For more information, see the *MKS Deploy 2009 Administration Guide*.

## Re-adding a Dropped Member

If you are re-adding a dropped member, an archive already exists for the member, and you must specify whether you want MKS Integrity to associate the member with the existing archive or generate a new one.

MKS Integrity only determines an archive exists for the member if a member with the same name previously existed at the location you are adding the member to, even though the backing archive may exist in another location.

The following cases also might cause you to be prompted to use an existing archive for the member you are adding:

- The member's project was migrated from a file system repository to a database repository.
- The dropped member was created by sharing another archive.
- The dropped member was renamed or moved before it was dropped.

## Create Archive Options (GUI)

Options	Description
<b>Author</b>	Your user name. The user name specified in the <b>Add Members</b> command configuration displays by default.
<b>Revision Number</b>	Revision number for the member. MKS Integrity creates the revision number 1.1 by default.
<b>Data Type</b>	File data type. The archive is stored in the specified format. The available options are: <ul style="list-style-type: none"> <li>▪ <b>Auto</b> means the file data type is determined automatically by MKS Integrity. For binary files less than 16K, specify the <b>Binary</b> file data type.</li> <li>▪ <b>Text</b> is the file format expected by text file editors.</li> <li>▪ <b>Binary</b> is file data type containing unprintable characters or lines too long to be handled by text editors.</li> </ul>
<b>On Existing Archive</b>	Options that apply in the event that MKS Integrity finds an existing archive for the member you propose to add: <ul style="list-style-type: none"> <li>▪ <b>Query</b> User asks you for confirmation on the action to be taken.</li> <li>▪ <b>Share Archive</b> uses the existing archive for the new member.</li> <li>▪ <b>Create New Archive</b> creates a new archive for the new member. MKS Integrity automatically generates the archive name and leaves the old archive unmodified.</li> <li>▪ <b>Cancel</b> cancels the operation.</li> </ul> <b>Note:</b> If you are replacing a text file with a symbolic link file of the same name, do not share the archive.
<b>Save Working File Timestamp</b>	To set the timestamp of the revision in the history to the timestamp of the working file, rather than the time the member was added, select <b>Save Working File Timestamp</b> .
<b>Close Change Package</b>	To close the associated change package, select <b>Close Change Package</b> . This option only displays if your administrator has enabled the use of change packages.

Options	Description
<b>Unexpand Keywords</b>	To replace literal values in the revision with keywords, select <b>Unexpand Keywords</b> .
<b>Defer Add</b>	To defer adding the member to the project, select <b>Defer Add</b> . This delays the add operation in the project until the deferred operation is submitted (see “Submitting Deferred Operations” on page 357). The operation in the Sandbox still takes place immediately. If change package reviews are mandatory, and this option is not selected, a pending member (and a pending change package entry) is created when the operation is complete. For more information, see “Change Package Reviews Overview” on page 417.
<b>Lock Revision</b>	Locks the revision using the lock type specified in your locks policy. For information on the locks policy, contact your administrator.
<b>Retain Working File</b>	To keep the working file of a newly added member, select <b>Retain Working File</b> .

## Create Archive Options (Web)

Options	Description
<b>Archive Description</b>	Description of the member.
<b>Change Package</b>	Change package that the operation is recorded in. This option only displays if your administrator has enabled the use of change packages.
<b>Deploy Type</b>	If the member is being added to a deploy project, specifies the deploy type for the member. This field only displays if you are licensed to use Deploy. For more information, see the <i>MKS Deploy 2009 Administration Guide</i> .
<b>Author</b>	Your user name. The user name specified in the <b>Add Members</b> command configuration displays by default.
<b>Revision Number</b>	Revision number for the member. The default revision number is 1 . 1.
<b>Data Type</b>	File data type. The archive is stored in the specified format. The available options are: <ul style="list-style-type: none"> <li>■ <b>Auto</b> means the file data type is determined automatically by MKS Integrity.</li> <li>■ <b>Text</b> is the file format expected by text file editors.</li> <li>■ <b>Binary</b> is a file data type containing unprintable characters or lines too long to be handled by text editors.</li> </ul>
<b>On Existing Archive</b>	Options that apply in the event that MKS Integrity finds an existing archive for the member you propose to add: <ul style="list-style-type: none"> <li>■ <b>Query User</b> asks you for confirmation on the action to be taken.</li> <li>■ <b>Share Archive</b> uses the existing archive for the new member.</li> <li>■ <b>Create New Archive</b> creates a new archive for the new member. MKS Integrity automatically generates the archive name and leaves the old archive unmodified.</li> <li>■ <b>Cancel</b> cancels the operation.</li> </ul> <p><b>Note:</b> If you are replacing a text file with a symbolic link file of the same name, do not share the archive.</p>
<b>Close Change Package</b>	In the field for <b>Close Change Package</b> , select one of the following options: <ul style="list-style-type: none"> <li>■ <b>Yes</b> closes the associated change package.</li> <li>■ <b>No</b> keeps the associated change package open.</li> <li>■ <b>Confirm</b> asks you for confirmation on the action to be taken.</li> </ul> <p>This option only displays if your administrator has enabled the use of change packages.</p>
<b>Lock Revision</b>	Locks the revision using the lock type specified in your locks policy. For information on the locks policy, contact your administrator.
<b>Unexpand Keywords</b>	To replace literal values in the revision with keywords, select <b>Unexpand Keywords</b> .

# Adding Dropped Members to a Project

---

**CLI EQUIVALENT** `si addmemberfromarchive`

---

Use the Add From Archive command to add a dropped member from an archive located on the server. You can also use this command to share the history of another member, where the source member can be located in another project from the added member.

The Add From Archive command:

- can be deferred when performed from a Sandbox.
- may be part of change package review (displayed as pending members in the project).
- supports multiple selection of archives using the wizard.

## To add members from archive in the GUI

- 1 Select the project, subproject, Sandbox or sub Sandbox you want to add members to.
- 2 Select **Member > Add From Archive**. The **Add Members From Archive Wizard** displays.
- 3 Members are added by specifying a member name with an archive location; together they define an entry that represents the destined member.

Do one of the following:

- Click **Select Archives** to browse the repository for an archive. The **Locate the Member Archive(s)** dialog box displays.

Select the archives corresponding to the members you want to add to the project. Archives are identified by the  icon. If you are using a file system repository, by default the files are located in the subdirectory named `rcs`. To add the selection as an entry in the list, click **OK**.

- In the first field, type the name that represents the member being added. Then type the archive location in the subsequent field (requires a valid location to an existing archive). The default archive location is the project selected in step 1. To add the entry to the list, click **Add**.

Repeat to add additional entries. Added entries appear as follows:

- **Member Name** displays the name of the member (may differ from the archive file name).
- **Archive Location** displays the absolute path of the archive and the archive file name.

To remove an entry from the list, click **Remove**. Repeat to remove additional entries.

- 4 To add the operation to a change package, from the **Change Package** list select a change package. To create a change package, click **Create**.

---

**NOTE** If you choose to create subprojects, the specified change package is used to record the Create Subproject operation.

---

- 5 To modify the **Add Members From Archive Wizard** options, click **Options**.
- 6 When you are finished specifying entries to add, click **Finish**. The **Add Member From Archive** dialog box displays.
- 7 The archive location displays in the **Archive Location** field. To change the location, type a new location in the field, or click **Locate Archive** to select it.
- 8 To modify the Add Member From Archive options, click **Options**.

- 
- 9 When you are finished, click **OK** to add the member to the project. To add all subsequent members specified in the **Add Members From Archive Wizard**, click **OK to All**. The member is added to the project.

## Add Member(s) From Archive Options

Options	Description
<b>Revision Number</b>	Specifies a revision for the added member ( <b>Add Member From Archive</b> dialog box only).
<b>Create Subprojects</b>	Creates subprojects for each subdirectory encountered when adding members ( <b>Add Members From Archive Wizard</b> only).
<b>Close Change Package</b>	Closes the associated change package after the operation is complete. This option only appears if your administrator has enabled the use of change packages.
<b>Defer Add From Archive</b>	Delays the add operations in the project until the deferred operation is submitted (only available for the <b>Add Member From Archive</b> dialog box when the operation is performed from the <b>Sandbox</b> view). If reviews are mandatory and this option is not selected, a pending member (and a pending change package entry) is created when the operation is completed.

# Character Encoding

MKS Integrity automatically decodes UTF-8 revision contents based on the presence of a byte order mark (BOM) in the file.

MKS Integrity supports UTF-8 encoded files in the following views:

- Annotated Revision
- MKS Visual Difference
- MKS Visual Merge

You can change the character encoding used in each view through a **Character Encoding** drop-down list on the toolbar. To make the list visible in the GUI, select **View > Character Encoding**. The list is always visible in the Web interface.

---

**NOTE** When check in operations are performed on the same member from different locales, ensure that either UTF-8 character encoding is used, or the characters used are standard ASCII. Otherwise, there may be unexpected results when opening the files in an editor or compiler configured for a different locale.

---

MKS Integrity supports the following encoding/decoding character sets:

Region	Character Set
Unicode	Unicode (UTF-8)
Western Character Sets	English (US-ASCII) Western (Windows-1252) Western (ISO-8859-1) Western (ISO-8859-15) Western (IBM-437) Western (IBM-850) Western (IBM-858) French-Canadian (IBM-863)
Japanese Character Sets	Japanese (EUC_JP) Japanese (ISO_2022_JP) Japanese (ISO_2022_JP_2) Japanese (Shift_JIS) Japanese (x-euc-jp-linux) Japanese (x-eucJP-Open) Japanese (x-windows-iso2022jp)

You can set character encoding preferences for each view and command through your client preferences. These preferences are used if the character set cannot be determined through the file's BOM, or if no character set is specified in the command line. By default, the character set in the preferences matches the default character set provided by the client's operating system locale.

---

**NOTE** The encoding settings do not apply to third party merge or differencing tools.

---

---

# Dropping Members From a Project

---

**CLI EQUIVALENT** `si drop`

---

If a member has outlived its usefulness or does not belong in a project anymore, you can remove it at any time.

In the GUI, you can remove a member through a **Sandbox** view or through a **Project** view.

In the Web interface, because Sandboxes do not exist, you can remove members through the **Project** view only.

After you remove a member from a project, the member is no longer listed as part of the Sandbox or master project, but the member's history remains in the project record, in case you need to recreate an earlier version of the project.

A *former member* is one that is dropped from the project, but still has a working file in the Sandbox. MKS Integrity retains the member history for former members as part of the project. Depending on the options you select when dropping the member, MKS Integrity can also delete the member's working file and close any associated change package.

---

**NOTE** You cannot check in a member that is associated with a deferred drop operation.

---

## To drop a member in the GUI

- 1 Select one or more members to remove.
  - 2 Select **Member > Drop**. The **Drop Member** dialog box displays.
  - 3 Under **Change Package**, select a change package, if applicable, or create a change package to link to.
  - 4 To modify the Drop Member options, click **Options**.
- Once a member is dropped from a project, the change is not immediately reflected in other Sandboxes. Other users must resynchronize their Sandboxes.
- 5 To drop the selected member, click **OK**. To drop multiple members, click **OK to All**. If you selected the **Confirm Drop** option, the **Confirm Drop** dialog box displays.
  - 6 To drop the selected member(s), click **Yes**. The project or Sandbox is updated to reflect the removed member, and the member is removed from the project.

---

**IMPORTANT** If change packages are enabled, the **Associate Change Package with dropped member** dialog box displays.

---

## To drop a member in the Web interface

- 1 From a **Project** view, select the member(s) you want to drop from the project by clicking the corresponding check box(es).
- 2 Select **Project > Drop Members**. The **Drop Member** dialog box displays.
- 3 Modify the **Drop Member** options as necessary.
- 4 To drop the selected member, click **OK**. To drop multiple members, click **OK to All**. If you selected the **Confirm Drop** option in step 3, the **Confirm Drop** dialog box displays.

- 
- 5 To drop the selected member(s), click **Yes**. The member is dropped from the project. The project is updated to reflect the dropped member.

**IMPORTANT** If change packages are enabled, the **Associate Change Package with dropped member** dialog box displays.

---

## Drop Member Options (GUI)

Options	Description
<b>Delete Working File</b>	Deletes the working files of the members being dropped.
<b>Confirm Drop</b>	Prompts you to confirm the drop for each selected member.
<b>Defer Drop Member</b>	Delays the drop operation in the project until the deferred operation is submitted. The operation in the Sandbox still takes place immediately. If change package reviews are mandatory, and this option is not selected, a pending member (and a pending change package entry) is created when the operation is complete.
<b>Close Change Package</b>	Closes any associated change package. This option displays only if your administrator has enabled the use of change packages.

## Drop Member Options (Web)

Options	Description
<b>Change Package</b>	Displays the associated change package, if any. This option only displays if your administrator has enabled the use of changed packages.
<b>Confirm Drop</b>	Prompts you to confirm the drop for each selected member.
<b>Close Change Package</b>	Closes any associated change package. Select one of the following options: ■ Yes closes the associated change package. ■ No keeps the associated change package open. ■ Confirm asks for confirmation about the action to be taken. This option displays only if your administrator has enabled the use of change packages.

---

# Filtering Members

For large software projects, use filters to display only the relevant members for your task. Filters allow you to view and manipulate a predefined subset of project members that are grouped according to their properties.

In the GUI, filters appear in the **Filter** list located on the local toolbar in the **Project** and **Sandbox** views. The software configuration management Web interface provides a reduced set of filters related to managing project members. The Web filters appear in the **Filter** list located on the toolbar.

You can select from the following filters:

Filter	Description
<b>All Members</b>	Shows all members of the current project or Sandbox.
<b>Sparse Sandbox Contents</b>	Shows only existing working files and deferred operations in a sparse Sandbox. This filter displays only if you have a sparse Sandbox open. A sparse Sandbox is a Sandbox that does not retain working files when a member is checked in.
<b>Any Member Deltas</b>	Shows only members that have working file changes or are out of sync with the member revisions in the project.
<b>Modified Working Files</b>	Shows only members with a working file that has been modified.
<b>Out of Sync Members</b>	Shows only members where the Sandbox revision is not the same as the member revision in the corresponding project.
<b>Out of Scope Members</b>	Shows only members that are outside of the scope definition of the Sandbox (if the Sandbox is a Scoped Sandbox). Specifying a Sandbox scope allows you to define what project members are included in a Sandbox, transferring specific members from the MKS Integrity Server to the Sandbox directory when the Sandbox is created and controlling what members display in the Sandbox view.
<b>New Revision Available</b>	Shows only those members where a new revision is available. If you apply the <b>New Revision Available</b> filter in a variant Sandbox, the filter returns only new revisions available in the variant Sandbox. It does not return new revisions from the master project.
<b>Working File Size Deltas</b>	Shows only members with changes in the size (bytes) of the working file.
<b>Missing Working Files</b>	Shows only those members with missing working files.
<b>Existing Working Files</b>	Shows only members with existing working files.
<b>New Members</b>	Shows only members that are newly added to the project and that have not yet been modified.
<b>Working File on Branch</b>	Shows only members where the working file is on a branch from a given development path that is not the main development path.
<b>Unresolved Merges</b>	Shows only members affected by unresolved merges.
<b>Work in Progress</b>	Shows only members that are considered work in progress. Combines the <b>Deferred Items</b> , <b>Members Locked By Me</b> , and <b>Modified Working Files</b> filters.
<b>Locked Members</b>	Shows only members locked by any user.
<b>Members Locked By Me</b>	Shows only members locked by you.
<b>Frozen Members</b>	Shows only frozen members. Changes to frozen members are not included as part of the member information in the project.
<b>Members with Attribute</b>	Shows only members with a particular attribute. If you choose this filter, MKS Integrity prompts you for the target attribute and value. In the GUI, enter the target attribute name in the <b>Attribute</b> field and the value for that attribute in the <b>Value</b> field, then click <b>OK</b> . In the Web interface, under <b>Specify a Member Attribute</b> , enter the attribute name in the field, then click <b>OK</b> . Enter the target attribute name in the <b>Attribute</b> field and the value for that attribute in the <b>Value</b> field, then click <b>OK</b> .
<b>Members at Label</b>	Shows only members that have the specified label assigned at the member revision. If you choose this filter, MKS Integrity prompts you for the target label. Enter the label name, then click <b>OK</b> .
<b>Members with Label</b>	Shows only members with the specified label somewhere in their member history. If you choose this filter, MKS Integrity prompts you for the target label. Enter the label name, then click <b>OK</b> .
<b>Members at State</b>	Shows only member revisions that have been assigned a specific promotion state. If you specify this filter, MKS Integrity prompts you to identify the target state. For more information on promotion states, contact your administrator. Enter a state, then click <b>OK</b> .

Filter	Description
<b>Members with Name</b>	Shows only members with the specified name. If you specify this filter, MKS Integrity prompts you to identify the target name. Enter a name, including the file extension, in the <b>Name</b> field (for example, utility.dll), then click <b>OK</b> .
<b>Members on Branch</b>	Shows only members that are off the main line of development.
<b>Pending Operations</b>	Shows any members in your Sandbox that are associated with a pending operation.
<b>Members With Rule</b>	Shows any members that have a member rule.
<b>Member Archive Shared</b>	Shows any members that share another member's archive. This filter displays only if you are using the database repository.
<b>Deferred Operations</b>	Shows any members in your Sandbox that are associated with a deferred operation, such as a deferred add, drop, checkin, move, or rename.

**NOTE** The list of built-in filters only displays if a **Project** or **Sandbox** view is active. You must expand your project (including subprojects) or your Sandbox (including sub Sandboxes) to view the filtered members.

By selecting the option for **Hide Empty Sandboxes** or **Hide Empty Projects**, you can remove any Sandboxes or projects that do not contain members matching the current filter. Directories that do not contain members are also removed. Selecting the **Hide Empty Sandboxes** or **Hide Empty Projects** filters causes MKS Integrity to search each subproject recursively for members matching the current filter and is therefore a client resource intensive operation. If you have a complex project with numerous members, you may want to limit your use of this filtering option.

Once a filter is applied to a project, operations performed on all project members apply only to those members shown as a result of the filter. For example, if you apply the filter for **Members at State** and then perform a **Lock Members** operation for that project, the lock operation applies only to those members shown by the **Members at State** filter.

You can also perform operations on a group of members by using the **Select** command.

# Importing Members into a Project

---

## CLI EQUIVALENT `si import`

---

When you add a member, it is automatically registered with the MKS Integrity Server; however, members added with Source Integrity Standard (an earlier version of MKS Integrity) or server resident files must be imported to register them with the MKS Integrity Server.

Do not use the Import Member command to add members that are dropped from a project. Instead, use the Add From Archive command.

---

**NOTE** The Import Member command is not supported for database repositories.

---

## Key Considerations

- The configuration management project must be registered with the MKS Integrity Server before the member is imported.
- Before you begin the import operation, the files to be imported must first reside on the MKS Integrity Server.
- Archives cannot be imported. When selecting a file to import, select the working file in the project directory. If the working file does not exist, select the project directory and enter the file name.
- You cannot import members into a shared subproject.

## To import a member in the GUI

- 1 From a **Project** or **Sandbox** view, select **Member > Import**. The **Import Members Wizard** displays.
- 2 To import a list of files to the project, click **Add File**. To import a directory and its contents, click **Add Directory**. The **Select One or More Members to Import to the Project** dialog box displays.
- 3 Select one or more files from the displayed list, navigating to the desired directory if necessary.

---

**NOTE** If your connection to the MKS Integrity Server is disconnected while you are browsing for a file, the file browser does not show any files or directories.

---

- 4 To add the operation to a change package, from the **Change Package** list select a change package. To create a change package, click **Create**.

---

**NOTE** If you choose to create subprojects, the specified change package is used to record the Create/Add Subproject operation.

---

- 5 Click **OK**. The files are added to the members list. To remove files, select the files, then click **Remove**.
- 6 To modify the Import Member wizard options, click **Options**.
- 7 Click **Finish** to import files without specifying options. The **Import Member** dialog box displays.
- 8 If the member is being imported to a deploy project, specify the deploy type for the member in the **Deploy Type** field. This field only displays if you are licensed to use Deploy. For more information, see the *MKS Deploy 2009 Administration Guide*.
- 9 To modify the Import Member options, click **Options**.

---

**10** To complete the operation, click **OK**. The members appear in the **Project** or **Sandbox** view.

## Import Members Wizard Options

Options	Description
<b>Recurse into Directories</b>	Imports members that exist in the specified directory location recursively.
<b>Create Subprojects</b>	Creates subprojects.
<b>Close Change Package</b>	Closes the change package. This option only appears if your administrator has enabled the use of change packages.

## Import Member Options

Options	Description
<b>Author</b>	Your user name. The user name specified in the command configuration displays by default.
<b>Data Type</b>	File data type. The archive is stored in the specified format. The available options are: <ul style="list-style-type: none"><li>■ <b>Auto</b>: file type determined automatically by MKS Integrity.</li><li>■ <b>Text</b>: file format expected by text file editors.</li><li>■ <b>Binary</b>: file containing unprintable characters or lines too long to be handled by text editors.</li></ul>
<b>Revision Number</b>	Revision number for the member. The default revision number is 1.1.
<b>Description</b>	Description of the member.
<b>Close Change Package</b>	Closes the change package. This option only appears if your administrator has enabled the use of change packages.
<b>Unexpand Keywords</b>	Replaces literal values in the revision with keywords.
<b>Defer Import</b>	Defers importing the member into the project (only available when the operation is performed from a <b>Sandbox</b> view). This delays the import operation in the project until the deferred operation is submitted. The operation in the Sandbox still takes place immediately. If change package reviews are mandatory, and this option is not selected, a pending member (and a pending change package entry) is created when the operation is complete.

# Viewing Member / Revision Information

**CLI EQUIVALENT** si memberinfo, si revisioninfo

The **Member Information** view displays information for a specified member. The **Revision Information** view displays information for a specified revision.

In the GUI, select **Member > Views > View Information**, or in the Web, select **Member > Member Information**. The **Member Information** view displays.

The **General** tab displays the following member information.

Field	Description
<b>Member Name</b>	Path and name of the member.
<b>Project/Sandbox Name</b>	Path and name of the member's project or Sandbox.
<b>Member Revision/Revision</b>	Revision number of the selected member or revision. In the <b>Member Information</b> view, you can select another revision by choosing a revision from the list.
<b>Archive shared</b>	Specifies if the member shares another member's archive. This option displays only if you are using the database repository.
<b>Member is a symbolic link (Member Information view only)</b>	Text displays only if the member is a symbolic link file.
<b>Created By</b>	Name of the user who created the revision and the date and time it was created.
<b>Locks</b>	The locks table lists all the locks on the revision. For each lock, it displays the following information by default: <ul style="list-style-type: none"><li>■ <b>Locker</b> displays the name of the user who locked the revision.</li><li>■ <b>Lock Type</b> displays whether the revision is locked with an exclusive or non-exclusive lock.</li><li>■ <b>Development Path</b> displays the name of the development path that the revision is locked on.</li></ul>
<b>State</b>	State assigned to the revision. To change the state of the revision, choose a state from the <b>State</b> list. States are defined by your administrator.
<b>Revision Description</b>	Brief description of the revision. You cannot change an existing revision description from this dialog box, but you can append additional comments to it. To do so, enter any supplemental information in the <b>Revision Description</b> field below the present information.

**NOTE** As part of the change package review process, if the member has a pending operation against it or one of its revisions, or if the member is located within a pending subproject, information displays in a note, for example, This member has a pending update to member revision 1.2 by devans.

The **Member Attributes** tab displays any attributes used to categorize member. You can use member attributes to perform operations on members as a group. For example, in the GUI you could use the **Select** command to highlight only those members with the attribute `sys=dos` and then check them out as a group. You can add or delete member attributes on this tab. Attributes conform to either of the following formats:

- *variable*

- 
- *variable=value*

---

**NOTE** No variable can exceed 80 characters, and no value can exceed 1024 characters.

The **Labels** tab displays displays, in alphabetical order, any labels used to describe and refer to a revision of a member. Labels can be based on the product release the revision was included in, on the content of the revision, on changes made to the revision, or any other sort of information that would be useful in identifying that particular revision. Although you generally assign a label to a new revision upon check in, you may sometimes want to add an additional label or change the label assigned to a revision.

The **Change Package** tab displays any change packages associated with the member or revision.

The **Member Revision** tab displays the project and development path that the member or revision is the member revision in. Only projects that you have the `OpenProject` permission for display.

The **Rules** tab displays for the Member Information View only. A member rule is a revision – typically a symbolic revision – attached to a member. You can use it with any command that allows you to specify a revision based on the member rule.

---

**NOTE** To edit a rule, you require the `ModifyMemberRule` permission.

# Selecting Members

The **Select** command allows you to highlight any members of the current Sandbox or project that meet specified selection criteria. Members selected with this command can be manipulated as a group by other MKS Integrity commands. This command allows you to instantly select a specific group of members in a project that might contain hundreds or thousands of members.

To select specific members, from a **Project** or **Sandbox** view, select **View > Select**.

You choose selection criteria by selecting a check box. To invert a filter, click the filter a second time and the ! symbol displays. For example, select members that are not frozen.

If two or more criteria are selected, they can be joined using the **Logical AND** or **Logical OR** option.

All options only apply to visible members. Members in unexpanded directories and subprojects are not selected.

Options	Description
<b>All Members</b>	Selects all visible members.
<b>Changed Members</b>	<p>Selects members that have specific changes to them.</p> <p>To choose all members that have changes, from the <b>Changed Members</b> list select Any Member Deltas.</p> <p>To choose all members that have working file changes, from the <b>Changed Members</b> list select Modified Working Files. Note that nothing is selected for this option if you perform the operation from the <b>Projects</b> view.</p> <p>To choose all members whose working file revision is different from the member revision, from the <b>Changed Members</b> list select Out of Sync Members. Note that nothing is selected for this option if you perform the operation from the <b>Projects</b> view.</p> <p>To choose all members that have a new revision available, from the <b>Changed Members</b> list select New Revision Available.</p> <p>To choose all members that have working file size changes, from the <b>Changed Members</b> list select Working File Size Deltas.</p> <p>To choose all members that are missing working files, from the <b>Changed Members</b> list select Missing Working Files.</p> <p>To choose all members that are new in the project, from the <b>Changed Members</b> list select New Members.</p> <p>Note: Selection criteria based on changes to working files will not select anything if this operation is performed from the <b>Projects</b> view.</p>
<b>Members Locked By</b>	<p>Selects members locked by a specific user.</p> <p>To choose members locked by you, select &lt;Me&gt; from the <b>Members Locked By</b> list.</p> <p>To choose members locked by any user, select &lt;Anyone&gt; from the <b>Members Locked By</b> list.</p> <p>To choose members locked by a specific user, select a user name, for example, &lt;mkern&gt;, from the <b>Members Locked By</b> list.</p> <p>If you select &lt;Me&gt; or &lt;Anyone&gt;, members where you have a lock on the working file only are included.</p>
<b>Member Lock Type</b>	Selects members locked with the specified type of lock.
<b>Members At Label</b>	<p>Selects members with a specific label. Select a label from the <b>Members With Label</b> list.</p> <p>To choose members with any label, select &lt;Any Label&gt; from the <b>Members With Label</b> list.</p> <p>Note: This option is not available if there are no members with labels in the currently selected project or Sandbox.</p>
<b>Members At State</b>	<p>Selects members with a specific state. Select a state from the <b>Members With State</b> list. States are defined by your administrator.</p> <p>Note: This option is not available if there are no members with states in the currently selected project or Sandbox.</p>

Options	Description
<b>Members With Attribute</b>	Selects members with a specific attribute. Select an attribute from the <b>Members With Attribute</b> list. Note: This option is not available if there are no members with attributes in the currently selected project or Sandbox.
<b>Deferred Items</b>	Selects members with a deferred operation associated with it. Select a deferred operation from the <b>Deferred Items</b> list. Note: This option is not available if there are no members with deferred operations in the currently selected project or Sandbox.
<b>Frozen Members</b>	Selects all frozen members (the revision and attributes cannot be updated).
<b>Members With Name</b>	Selects members with a specific name. Enter a member name in the <b>Members With Name</b> field. Wildcards such as ? and * are supported.
<b>Working revision branched from Member Revision</b>	Selects members where the working file is on a branch from a given development path that is not the trunk development path.
<b>Members on Branch</b>	Selects only members that are off the main development trunk.
<b>Out of Scope Members</b>	Selects only members with working files in the Sandbox that are outside of the scope definition of the Sandbox (if the Sandbox is a Scoped Sandbox). Specifying a Sandbox scope allows you to define what project members are included in a Sandbox, transferring specific members from the MKS Integrity Server to the Sandbox directory when the Sandbox is created and controlling what members display in the Sandbox view.
<b>Pending Operations</b>	Selects members with pending operations associated with them. Select the pending operation from the <b>Pending Operations</b> list.
<b>Members With Rule</b>	Selects members based on one of the following revision rule filters: <b>Members With Rule</b> selects all members with a revision rule. <b>Member Revision Differs From Rule</b> selects all members for which the rule does not match the member revision. <b>Invalid Rule</b> selects all members for which the rule does not expand to any existing revision.

---

# Moving Members Between Projects

---

**CLI EQUIVALENT** `si move`

---

As software designs and project structures change, it may become necessary to move one or more members between projects, variants of the same project, or directories in a project.

For example, if you are re-designing and expanding your company web site, you can organize the site more efficiently by moving some of the files to new and existing directories and subprojects.

Moving a member performs a drop in the source project and an add in the target project, creating a new revision in the existing archive. The member's archive remains in its original location in the repository so that change packages, member histories, and project histories continue to work. If the move is performed with a change package, the member is added as a single `Move` entry in the change package. If you are moving multiple members, any common directory prefix shared by the members is automatically removed by default during the move.

You can move one or more members between projects or directories in a **Project** or **Sandbox** view using the menu command or by using the drag-and-drop method.

## **Key Considerations**

- Moving members between projects on different servers is not supported.
- To move a member, you require the `DropMember` permission on the project you are moving the member from and the `AddMember` permission on the project you are moving the member to.
- You can perform deferred member moves only if both the source and target locations are Sandboxes.
- The `move members` command does not work recursively on subprojects. To move one or more subprojects, use the **Project > Subproject > Move** command.
- MKS Integrity detects whether any ACLs exist in the tree defined by the source project, the target project, and the common root between them (or the global ACL, if there is no common root). If any ACLs are found, MKS Integrity warns you so you can manually make any required adjustments to the ACLs after the move is complete. For more information on ACLs, see the *MKS Integrity Server 2009 Administration Guide*.

## To move a member in the GUI

- 1 Select one or more members to move.
- 2 Select **Member > Move**. The **Move Members Wizard** displays.

---

**NOTE** Depending on the view you performed the move from, the **Project** or **Sandbox** option is selected and the name of the project or Sandbox displays.

---

- 3 Specify the destination project or Sandbox for the member(s):
  - To move the member(s) to a specific project, proceed to step 4.
  - To move the member(s) to a specific Sandbox, click **Sandbox**, then click **Select** to choose a Sandbox from the list. Proceed to step 8.

---

**NOTE** If you specify a Sandbox, the move occurs in the corresponding master project.

---

- 
- 4 Click **Project**, then click **Select**. If you are performing the move operation from a **Project** view, proceed to step 7. If you are performing the move operation from a **Sandbox** view, the first panel of the **Specify the target project for the move wizard** displays.
  - 5 Enter the path and name of the project and proceed to step 10, or click **Select** to choose a project from the list.

---

**NOTE** If you are moving the member(s) to a variant or build subproject, only enter the path and name of the root project in this field. You specify the subproject later in the procedure.

---

- 6 Click **OK**. The second panel of the **Specify the target project for the move wizard** displays.
- 7 Select the type of project you want to move the member(s) to by clicking a project type option. The available types are:
  - **Normal** moves the member (s) to a project based upon the current state of the project. Click **Finish** and proceed to step 10.
  - **Variant** moves the member(s) to a project based upon a specific development path.  
From the **Development Path Name** list, select a development path name, for example, `Aurora_Beta_Variant`.

---

**NOTE** The **Variant** option is unavailable if there are no available development paths.

---

- 8 If you are moving the member(s) to a variant or build project, click **Next** to select a subproject. The third panel of the **Specify the target project for the move wizard** displays.
- 9 Expand the project to select the specific subproject that you want to move the member(s) to. Click **Finish**.

---

**NOTE** There are rules that control what project configuration you can jump to. If your selection breaks any of the rules, you cannot move the member.

---

- 10 Under **Change Package**, select a change package, if applicable, or create a change package to link to.

---

**NOTE** If you choose to create subprojects, the specified change package is used to record the Create Subproject operation.

---

- 11 To modify the Move Members options, click **Options**.
- 12 Click **Next**. The second panel of the **Move Members Wizard** displays.

---

**NOTE** The options that appear depend on whether you selected a destination project or destination Sandbox.

---

---

**13** Specify the destination directory for the member(s):

- To move the member(s) to the destination project's directory, click **Destination Project Directory**.
- To move the member(s) to the destination Sandbox's directory, click **Destination Sandbox Directory**.
- To move the member(s) to a subdirectory in the destination project's directory, click **Subdirectory of Destination Project Directory**, then type the subdirectory, for example, bin/etc.
- To move the member(s) to a subdirectory in the destination Sandbox's directory, click **Subdirectory of Destination Sandbox Directory**, then type the subdirectory, for example, bin/etc.

**14** Click **Next**. The third panel of the **Move Members Wizard** displays, summarizing the details of the move.

**15** Review the settings you selected and make changes if necessary.

**16** To move the selected member, click **Finish**. If you enabled the **Confirm Move** option, MKS Integrity prompts you to confirm the move.

**17** To confirm moving the member, click **OK**. (for multiple members, click **OK to All**). The member(s) appear in the new location.

### To move a member by dragging in the GUI

You can move one or more members by left-clicking and dragging one of the following:

- One or more member nodes.
- A directory node. This automatically moves all members in the directory and subdirectories. If any subprojects or sub Sandboxes are encountered, this initiates the **Move Subproject Wizard**, prompting you to confirm the move.
- A subproject or sub Sandbox node. This initiates the **Move Subproject Wizard**, prompting you to confirm the move.

You can drag a member onto a project, subproject, Sandbox, sub Sandbox, or directory node in the active **Project** or **Sandbox** view, or an adjacent open **Project** or **Sandbox** view. The drag-and-drop action initiates the **Move Member** wizard, prompting you to confirm the operation.

## Move Members Options

Options	Description
<b>Move Working File</b>	Moves the working file into the Sandbox immediately. <b>Note:</b> This option is only valid if you are moving one or more members from a source Sandbox to a target Sandbox
<b>Overwrite Existing File</b>	Overwrites the target (new name) working file if it exists. <b>Note:</b> This option is valid only if you are moving one or more members from a source Sandbox to a target Sandbox.
<b>Confirm Move</b>	Confirms the move before proceeding.
<b>Defer Move</b>	Delays the move operation in the project until the deferred operation is submitted. <b>Note:</b> This option is valid only if you are deferring the move of one or more members from a source Sandbox to a target Sandbox.
<b>Create Subprojects</b>	Creates a subproject in the directory that the member is moving to (if it does not exist) and adds the member to it.
<b>Create Branch if Variant (GUI only)</b>	Creates a branch for the moved member in the target location.
<b>Close Change Package</b>	Closes the associated change package. This option only appears if your administrator has enabled the use of change packages.

# Identifying Members to Add to Projects

From MKS Integrity, you can view non-member files (files that have not been placed under version control) residing in your Sandbox directory. The **Non-Members** view is useful when used recursively to identify all of the files that may need to be placed under source control.

To view non-members, from a **Sandbox** view, select **Sandbox > Views > View Non-Members**.

---

#### NOTE

- If a member is selected when you display the **Non-Members** view, the non-members for the nearest project or subproject are displayed. The project or subproject name used by the **Non-Members** view displays in the title bar.
  - You cannot display multiple **Non-Members** views for the same Sandbox.
- 

The **Non-Members** view does not display files that are:

- deferred imported members
- deferred add members from archive
- pending members (add, add from archive, import)
- working files from members that are renamed but not resynchronized

By default, former members (members that were dropped from your Sandbox) are not displayed in the **Non-Members** view.

You can add a non-member to a project, or delete or edit a non-member by selecting one or more files and selecting the appropriate option on the **View** menu. You can also filter the contents of the **Non-Members** view.

#### **Filtering Non-Members**

You can filter the content of the **Non-Members** view by specifying files, file types, or directories to include or exclude.

To filter the view, do one of the following:

- Select one or more files and select **View > Exclude with Name** or **View > Exclude with Extension**.

---

**NOTE** Files excluded by name are not excluded once the **Non-Members** view is closed.

---

- Select **View > Filters** and specify any combination of files or directories to include or exclude. By specifying file types to include, all file types not specified are excluded.

---

**NOTE** If filters are specified in the **Non-Members** view preferences, those filter settings appear in the **Non-Members Filter** dialog box.

---

---

**TIP** To specify all files with a specific file extension, use the wildcard (\*), for example, \*.java.

---

---

## Non-Members View Information

Column	Description
<b>Closest Project</b>	Project or subproject, with file path, corresponding to the Sandbox that is closest to the directory containing the file.
<b>Closest Sandbox</b>	Sandbox or sub Sandbox, with file path, that is closest to the directory containing the file.
<b>Member ID</b>	Default member name for the file as it would appear if it was added to the nearest project. In the case where the nearest project is a subproject, the relative path displays with the member name.
<b>Absolute Path</b>	Absolute file path of the file.
<b>Size</b>	Size of the file in bytes.
<b>Last Modified</b>	Date that the file was last modified.
Symbolic Link	File is a symbolic link file.

---

# Viewing and Editing Member Content

You can open a member in your default editor or in the editor associated with the file's extension. You configure your default editor using MKS Integrity Client preferences. In the Web interface, the working file member is opened in a second browser window.

You can view the contents of a member, or you can edit the working file for the member.

To view the contents of a member, in the GUI, select **Member > Revision > View Contents** or in the Web, select **History > View Revision**. When you view the contents of a member, MKS Integrity copies it to a read-only temporary file and opens it for you. The temporary file is not the revision. If you make changes to the file and want to save it, the actual revision is not modified.

To edit the contents of a member's working file, in the GUI, select **Member > Edit Working File** or in the Web, select **Member > View Member**.

When you edit a working file, whether you can edit or just view the contents of the member depends upon whether you have read-only or read-write access to the member. You must have a member locked in your name to be able to check it in.

---

**TIP** If you know the location of a revision, you can view it by typing the following URL in a browser:

`http://<server>:<port>/si/viewrevision?projectName=<(sub)projectname>&revision=<revisionnumber>&selection=<membername>`

for example

`http://xyzBusiness:7001/si/viewproject?projectName=c:/master_projects/SourceCode/frameworkproject.pj&revision=1.4&selection=setup.ksh`

---

## Scanning for Changes

Changes to member information are automatically displayed in the GUI in the **Sandbox** view. You can also use the **Scan for Changes** feature to display changes if the client is offline from the MKS Integrity Server or if you need to see changes to working file information on your local drive.

To scan a Sandbox for changes in the GUI, from a **Sandbox** view, select **View > Refresh**. The **Sandbox** view displays members whose working file has changed (indicated by working file deltas), or a message informing you there are newer revisions available in the member history (indicated by member deltas).

# Checking Out a Member

---

## CLI EQUIVALENT `si co`

---

Before you can make changes to a member, you must first check it out. When you check out a revision, it is copied to a working file where you can view or modify its contents.

*Checking out* a member extracts the contents of a revision in a member history and copies it to the working file. You can check out any revision by specifying either its revision number or label.

If you already have a member checked out, and you have not made any changes to the member, MKS Integrity does not perform the checkout a second time. While the checkout dialog box does appear, the original checkout stands.

In the configuration management Web interface, when you check out a revision, it is downloaded to a working file for editing. This occurs because Sandboxes do not exist in the Web interface.

## To check out a member in the GUI

- 1 From a **Sandbox** view, select one or more members to check out.
- 2 Select **Member > Check Out**. The **Check Out** dialog box displays.
- 3 Click the desired tab, then modify the checkout options. For additional checkout options, click **Options**.
- 4 Under **Change Package**, select a change package, if applicable, or create a change package to link to.
- 5 To check out a single selected member, click **OK**. To check out all selected members, click **OK to All**.

If you have made changes to your working file, and then attempt to check out the member, the **Confirm Overwrite Working File** dialog box displays. If you want to retain your changes in the working file, click **No (No to All** for multiple members). If you want to compare your working file with the revision you are checking out, click **Differences**. To proceed with the check out operation for a member, click **Yes** (for multiple members, click **Yes to All**).

If your working file is based on a different revision, you can merge it with the revision you are checking out.

The member is checked out for editing, indicated by a padlock icon, the locker's name, and the date and time of the lock.

The padlock icon indicates the type of lock and any potential conflicts with other lockers. For a description of the padlock icons, see the *MKS Integrity Client 2009 Getting Started Guide*.

## To check out a member in the Web interface

- 1 From a **Project** or **Member History** view, select a member to check out by clicking the corresponding check box.

---

**NOTE** In the Web interface, you can check out only one member per operation. Multiple member selections for checkout cause an error.

---

- 
- 2** From the **Project** view, select **Member > Check Out**. From the **Member History** view, select **History > Check Out**. The **Check Out** dialog box displays.

---

**NOTE** The **Change Package** options appear only if change packages are enabled.

---

- 3** Click the desired tab, then modify the Check Out options.
- 4** To check out the member, click **OK**. The **File Download** dialog box displays.

---

**NOTE** Once you reach the file download stage of the checkout operation, if you cancel the file download the member appears locked in the project. This occurs because the member is checked out and saved to a temporary location pending your download options.

To unlock the member, if you cancelled the file download, select the member by clicking the corresponding check box and then select **Member > Unlock**.

---

- 5** To download the member, select one of the following options:
- To open the member in its associated program immediately, select **Open this file from its current location** and click **OK**. The file is automatically sent to a temporary directory on your system. If the member you are checking out is a program or application (executable format) the **File Download** dialog box reappears with the following options:
    - To run the program immediately, select **Run this program from its current location** and click **OK**.
    - To save the program to a specified location on your local drive, select **Save this Program to disk** and click **OK**. The **Save As** dialog box displays. Specify a location for the program and click **Save**.
    - To save the member to a specified location on your local drive, select **Save this file to disk** and click **OK**. The **Save As** dialog box displays. Specify a location for the member and click **Save**.

The member is checked out for editing, indicated by a padlock icon, the locker's name, and the date and time of the lock.

The padlock icon indicates the type of lock and any potential conflicts with other lockers.

## Check Out Options

Check Out Dialog Box Tab	Revision to Check Out Options
<b>Selection</b>	<p>Specifies which revision of the member to check out.</p> <p>To check out a pre-defined revision, click <b>Pre-Defined Revision</b>, then select a revision type from the list. The available options are:</p> <ul style="list-style-type: none"> <li>■ Member checks out the member revision, that is, the revision shown in the <b>Project</b> view (this is the default).</li> <li>■ Working checks out the working file.</li> <li>■ Head checks out the head revision.</li> <li>■ Trunk Tip checks out the latest revision in the trunk.</li> <li>■ Member Branch Tip checks out the latest revision along the member's current branch of development.</li> <li>■ Member Rule checks out the revision defined in the member rule.</li> </ul> <p>To check out the latest revision, select <b>Specific Revision</b>. The <b>Specific Revision</b> option selects the most recent revision by default and displays it in brackets, for example, (1.1).</p> <p>To check out a revision by state, click <b>Latest revision with State</b>, then select a state from the list. The options in the list depend on the states configured by your administrator. This option is only available when promotion is enabled by your administrator and is only visible in the GUI.</p>
<b>Revisions</b>	Allows you to select a specific revision to check out. Select a revision from the member history.
<b>Properties (GUI only)</b>	<p>The <b>Properties</b> tab allows you to check out a revision with a specific state, label, or timestamp.</p> <ul style="list-style-type: none"> <li>■ <b>Last revision with state</b> allows you to check out a revision with a specific state. States are defined by your administrator.</li> <li>■ <b>Revision with label</b> allows you to check out a specific labeled revision.</li> <li>■ <b>Last revision at date</b> allows you to check out the revision closest to the specified timestamp. Because the revision closest to the specified timestamp may be on a different branch, you can limit the matching to a <b>Member Branch</b>, <b>Any Branch</b>, or a specific branch in the list.</li> </ul>
<b>Labels (Web interface only)</b>	Allows you to select a revision to check out by label. Select a label from the label list.
<b>Link (GUI only)</b>	<p>The <b>Link</b> tab allows you to check out a member in a specific project configuration (normal, variant, build). To link the project that this member belongs to (the target project) with the master project, configure the following options:</p> <p>To specify the master project, click <b>Member Revision in a Specified Project</b>. Next, click <b>Select</b> to browse for the specific project and configuration type.</p> <p>Under <b>Member</b>, choose an option to specify the member in the target project:</p> <ul style="list-style-type: none"> <li>■ <b>Find member in a specified project</b> searches for the member with the same backing archive in the target project.</li> <li>■ <b>Find member recursively</b> searches recursively throughout the subprojects for the member. There must be exactly one backing archive for each member.</li> <li>■ <b>Specify member name</b> is the name of the member in the target project. This is useful when the target project contains several members that point to the same archive as the current member.</li> </ul>

---

Check Out Dialog Box Tab	Revision to Check Out Options
<b>General</b>	<p>Specifies the following general checkout options:</p> <p><b>Lock Revision</b> allows you to determine whether the member is locked on checkout. You can select from the following options:</p> <ul style="list-style-type: none"> <li>■ To obtain a lock on checkout, select <b>Lock</b>.</li> <li>■ To checkout without a lock, select <b>No Lock</b>.</li> <li>■ To obtain a lock on checkout based on the locks policy, select <b>Follow Policy</b>. For information on the locks policy, contact your administrator.</li> </ul> <p><b>Move My Lock</b> moves any lock you have on a revision in the same development path to the member revision, including the change package associated with the lock operation. Since you can only have one lock per member per development path, if you already have another revision locked, you need to move that lock to the member revision in order for the check in to succeed. See the <b>Downgrade Lock Request on Existing Exclusive Lock</b> option on the <b>Advanced</b> tab for information on what occurs if the member revision already has an exclusive lock. You also need to move your lock if it is associated with a different change package than the one you are using for the check out operation.</p> <p><b>Overwrite if Deferred Operation Exists</b> (GUI only) overwrites the working file if the file has changed and there is a deferred pending operation on the member.</p> <p><b>Overwrite Working File if Changed</b> (GUI only) overwrites the working file even if it is modified.</p> <p><b>Change Package</b> specifies the change package that the operation will be recorded in.</p> <p><b>Line Terminator</b> (Web interface only) determines the type of line terminator MKS Integrity uses when dealing with members: <b>lf</b> (or line feed, primarily for UNIX applications), <b>cr</b>, or <b>crlf</b>.</p>

---

Check Out Dialog Box Tab	Revision to Check Out Options
<b>Advanced</b>	Specifies the following advanced checkout options:
	<b>Lock Type</b> allows you to determine the type of lock obtained on checkout. Select one of the following options:
	<ul style="list-style-type: none"> <li>■ To obtain an exclusive lock on the member, select <b>Exclusive</b>. An exclusive locks prevents other users from checking in changes to the member.</li> <li>■ To obtain a non-exclusive lock on the member, select <b>Non-Exclusive</b>. A non-exclusive locks enables other users to check in changes to the member.</li> <li>■ To obtain a lock type based on the locks policy, select <b>Follow Policy</b>. For information on the locks policy, contact your administrator. If the locks policy is <b>None</b> (no lock required), but the <b>Lock Revision</b> option is selected, a non-exclusive lock is obtained.</li> </ul>
	<b>Update Member Revision</b> causes the revision you check out to become the new member revision of the project. For example, if the current project member is listed as Revision 2.3 and you check out Revision 1.7 with the <b>Update Member Revision</b> option selected, Revision 1.7 replaces Revision 2.3 as the member revision of the project.
	<b>Downgrade Lock Request on Existing Exclusive Lock</b> obtains a non-exclusive lock when you try to get an exclusive lock on a revision that another user has an exclusive lock on.
	<b>Restore Revision Timestamp</b> (GUI only) sets the timestamp of the working file, to which the revision is checked out, to the date and time of the revision in the history. If this option is not set, the working file's timestamp is set to the current date and time.
	<b>Merge Working File if Changed</b> (GUI only) automatically merges any changes from the revision being checked out into the working file (if your working file is based on a different revision).
	<b>Keywords</b> allows you to select keyword expansion options when checking out a member.
	<ul style="list-style-type: none"> <li>■ To leave keywords as they are recorded in the revision, select <b>Do Not Expand</b> from the <b>Keywords</b> list.</li> <li>■ To replace keywords in the revision with literal values in the working file, select <b>Expand</b> from the <b>Keywords</b> list.</li> <li>■ To unexpand keywords in the working file, select <b>Unexpand</b> from the <b>Keywords</b> list.</li> </ul>
	<b>Merge Type</b> (GUI only) specifies the action to be taken when merging the checked out revision into the working file. Select one of the following options from the list:
	<ul style="list-style-type: none"> <li>■ <b>Confirm</b> confirms the action to be taken when merging upon checkout.</li> <li>■ <b>Cancel</b> cancels the operation.</li> <li>■ <b>Automatic</b> performs an automatic merge.</li> <li>■ <b>Manual</b> initiates the MKS Visual Merge tool.</li> </ul>
	<b>On Conflicts</b> (GUI only) specifies the action to be taken when merge conflicts occur. Select one of the following options from the list:
	<ul style="list-style-type: none"> <li>■ <b>Confirm</b> confirms the action to be taken when a conflict occurs.</li> <li>■ <b>Cancel</b> cancels the operation when a conflict occurs.</li> <li>■ <b>Mark For Later Merge</b> marks the files for merging at another time, allowing you to resolve the conflict first.</li> <li>■ <b>Launch Tool</b> initiates the MKS Visual Merge tool.</li> <li>■ <b>Highlight Output File</b> highlights conflicts in the resulting merged revision.</li> <li>■ <b>Error</b> displays an error message prompt.</li> </ul>

# Checking In a Member

---

**CLI EQUIVALENT** `si ci`

---

When you are satisfied with the changes you made to a member, you should check in the member to preserve those changes as a new revision in the member's history. Members should be checked in on a regular basis.

*Checking in* a member creates a new revision of a member and adds it to the member history. When a member is checked in to a revision other than the head revision or a branch tip revision, a new branch is created.

Interface	Procedure
GUI	From a <b>Sandbox</b> view, select one or more members to check in, and then select <b>Member &gt; Check In</b> .
Web	From a <b>Project</b> or <b>Member History</b> view, select a member to check in by clicking the corresponding check box. From the <b>Project</b> view, select <b>Member &gt; Check In</b> . From the <b>Member History</b> view, select <b>History &gt; Check In</b> . <b>Notes:</b> <ul style="list-style-type: none"><li>■ You can check in only one member per operation in the Web interface. Multiple member selections for check in cause an error.</li><li>■ You cannot check in a symbolic link file member in the Web interface. If you check in a symbolic link file in the Web interface, the link file is replaced with the contents of the target file.</li></ul>

---

**NOTE** If you are using a file system repository, the maximum size for members is 2 GB; if you are using a database repository, the maximum size is 2 GB for Oracle or SQL Server, and 1 GB for DB2. To find out the type of repository used on your server, see your administrator.

---

In the GUI, you can click the **Differences** button on the **Check In** dialog box to view the differences between your working file and the revision you checked out.

## Resynchronizing Before Checking In

If you use a non-exclusive locking policy, other users besides yourself may have the same revision checked out. If another user checks in their changes first, they get the next sequential revision on the current branch (if the revision is the tip of the current branch). When you try to check in your changes, you are prompted to resync and merge the previously committed changes into your working file before you can complete the check in. For more information on your locks policy, contact your administrator.

## Assigning Revision Descriptions

A *revision description* is text that becomes a permanent part of the archive's metadata. It allows you to provide a record of the changes you made and why you made them. This can be of great value to you or other team members if it ever becomes necessary to revise or update the member.

Once a new revision is checked in, its revision description cannot be changed. You can, however, append new information to a revision description.

If your administrator has set the feature for enforced revision descriptions, you must enter a revision description.

---

## **Using Change Packages**

If change package reviews are mandatory, a pending revision is created when the change package is submitted.

### **Updating the Member Revision**

You can use the **Update Member Revision** option when you are checking in a member to ensure the most recent revision of each member is added to the list of members in the project. If the option is not enabled, the project list might not reflect the most current revision of each member's history. For example, if the current project member is revision 2.7 of an archived file, but a newer revision 2.8 has been added to the member's history, you can update the member to the new revision.

### **Assigning Revision Numbers**

By default, when you check in a member, MKS Integrity automatically assigns a unique revision number to the new revision. It does this by incrementing the current revision number by one. For example, if the previous revision is 1.3, the new revision is assigned number 1.4. For best practices, MKS recommends letting MKS Integrity generate revision numbers for you—do not use revision numbers to record milestones (use labels or checkpoints). The ability to specify a specific revision number is supported for legacy purposes, but is not recommended.

You can choose the revision number of the changes you are checking in, so long as your revision number:

- is greater than the last revision number (you cannot use previously “skipped” revision numbers)
- has no leading zeros (zeros as complete revision numbers are acceptable)
- starts a new branch based on an existing revision (optional)

If you check in a revision using an already existing revision number, MKS Integrity attempts to add one to the revision number and check it in as that revision. If that revision already exists, MKS Integrity then chooses the next available branch number and creates a new branch.

For example, if you are checking in a new revision to an archive where the head revision is 1.7, the following numbers are valid:

- 1.8 (greater than head revision)—if you check in a revision as 1.7, which already exists, MKS Integrity assigns it 1.8
- 1.10 (greater than head revision)
- 1.72 (none of the numbers between 7 and 72 may be used afterwards)
- 2.0
- 1.7.1.1 (if it starts a new branch)
- 1.7.0.1 (leading zero as the branch number)

The following numbers are invalid:

- 1.3 even if there was no revision 1.3 previously (MKS Integrity branches the archive and assigns 1.3.x.1, where x is the first available branch number)
- 1.08 (leading 0 in last portion)
- 02.1 is considered the same as 2.1 (leading zero in branch number)

---

### **Starting a Branch When Checking In a Member**

MKS Integrity usually places a new revision at the top of the main trunk. There are times, however, when you do not want your work to be checked into the trunk. You may be pursuing a line of development that will not be included in the finished product, or you may be doing post-release maintenance while development for the next release continues on the trunk. Divergent lines of development in the same archive are managed through the use of branches.

### **Deferring a Check In**

In some situations, you cannot check in a member. For example, you cannot check in a member if:

- there are revision conflicts on the member that you do not yet want to resynchronize
- there is an exclusive lock held by another user on the member revision and you do not want to force the creation of a new branch
- the member revision is frozen

When you cannot check in a member, but still want to record information about the check in (for example, the revision description), you can defer the check in.

When a revision conflict occurs on a check in, a deferred check in is created automatically. In other situations, you can select the **Defer Check In** option.

### **Maximum Size of Text Working Files in Store-By-Delta Archives**

If you are using the database repository, your administrator may set the policy that determines the maximum size of text working files that can be stored in a store-by-delta archive. The policy is useful in preventing out of memory issues on the MKS Integrity Server when MKS Integrity attempts to difference large text files. If the file exceeds the maximum size during a check in, you are prompted to cancel the operation or convert the store-by-delta archive to store-by-reference.

### **Specifying a Source File in the Web Interface**

In the configuration management Web interface, because Sandboxes do not exist, you specify a source file (working file) for the member, which is checked in as the new revision.

If the name of the source file you specify is different than the member name, depending on how you have the **Different Member/Source File Name** option set, one of the following occurs:

- MKS Integrity confirms if you want to proceed. Click **Yes** if you want to check in a source file different from the member.
- MKS Integrity cancels the checkin operation because the file names do not match.
- The member is checked in.

## Check In Options

Check In Dialog Box	Options
General	<p>Specifies the following general checkin options:</p> <p><b>Label</b> is a unique string that identifies the new revision. Revision labels are usually assigned during check in, but can be assigned later, for instance, using the <b>Member &gt; Properties &gt; Add Label</b> command.</p> <p><b>Move Existing Label</b> moves the label if it already exists on another revision. In the Web interface only, select from one of the following options:</p> <ul style="list-style-type: none"><li>■ Select <b>Yes</b> to move the label.</li><li>■ Select <b>No</b> to not move the label.</li><li>■ Select <b>Confirm</b> to be asked for confirmation of the action to be taken.</li></ul> <p><b>Defer Check In</b> delays the checkin of the revision. If you have locked the member, your lock remains on the revision and MKS Integrity displays version information for both the working and member revisions. If a change package is specified, a <b>Deferred Check In (Lock)</b> or <b>Deferred Check In (No Lock)</b> entry is recorded in the change package. Once you submit the checkin, your lock is removed and the member revision moves to the next number in the sequence, as in the case of a standard checkin operation.</p> <p><b>Lock Revision</b> checks in the working file, then immediately lock the new revision. This allows you to update the archive while retaining control of the revision. The type of lock used is the same as the lock type used when the file was checked out.</p> <p><b>Check In if Unchanged</b> checks in the working file even if it has not changed since it was last checked out.</p> <p><b>Close Change Package</b> closes the change package associated with the member, and if the MKS Integrity integration is enabled, updates the item status in MKS Integrity. In the Web interface only, select from one of the following options:</p> <ul style="list-style-type: none"><li>■ Select <b>Yes</b> to close the associated change package.</li><li>■ Select <b>No</b> to keep the associated change package open.</li><li>■ Select <b>Confirm</b> to be asked for confirmation of the action to be taken.</li></ul> <p>This option only appears if your administrator has enabled the use of change packages.</p> <p><b>Update Member Revision</b> makes the new revision the member revision in the project, replacing the existing member revision.</p> <p><b>Different Member/Source File Name</b> (Web interface only) confirms the action to be taken if the specified source file name and member name are different.</p> <ul style="list-style-type: none"><li>■ Select <b>Yes</b> to allow the different file names.</li><li>■ Select <b>No</b> to disallow the different file names.</li><li>■ Select <b>Confirm</b> to be asked for confirmation of the action to be taken.</li></ul>

Check In Dialog Box	Options
<b>Advanced</b>	<p>Specifies the following advanced checkin options:</p> <p><b>Author</b> is the author name applied to the revision. If necessary, enter a name. The user name specified during the <b>Add Members</b> command displays by default.</p> <p><b>Revision Number</b> specifies the revision number you want to assign to the revision. By default, MKS Integrity creates the next logical revision number, for example, 1.1 to 1.2. Optionally, you can enter a revision number.</p> <p><b>Update Member Revision Even if not Current</b> causes MKS Integrity to update the member revision upon checkin, even when the revision being checked in is not the member revision.</p> <p>In the Web interface only, select from one of the following options:</p> <ul style="list-style-type: none"> <li>■ Select <b>Yes</b> to update the member revision.</li> <li>■ Select <b>No</b> to not update the member revision.</li> <li>■ Select <b>Confirm</b> to be asked for confirmation of the action to be taken.</li> </ul> <p><b>Force Creation of New Branch</b> creates a branch off the revision you are checking in.</p> <p><b>Note:</b> If required, a branch is created even if you do not select the <b>Force Creation of New Branch</b> option, for example, if you are checking in a member to a revision other than the head revision.</p> <p><b>Unexpand Keywords</b> controls whether keywords are expanded prior to being checked in to the archive.</p> <p><b>Save Working File Timestamp</b> (GUI only) sets the timestamp of the revision in the history to the timestamp of the working file, rather than the time of check in.</p> <p><b>Create Branch if Variant</b> creates a branch off the revision you are checking in, if you are working in variant Sandbox and this is the first time the member is checked in.</p> <p>In the Web interface only, select from one of the following options:</p> <ul style="list-style-type: none"> <li>■ Select <b>Yes</b> to create a branch.</li> <li>■ Select <b>No</b> to not create a branch.</li> <li>■ Select <b>Confirm</b> to be asked for confirmation of the action to be taken.</li> </ul> <p><b>Retain Working File</b> (GUI only) checks in the working file and immediately resynchronize it. If this option is cleared, the working file is deleted after it is checked in.</p> <p><b>On Newer Revision</b> allows you to determine what happens when the revision being checked in is not the member revision in the development path.</p> <p>You can select one of the following options:</p> <ul style="list-style-type: none"> <li>■ To resynchronize and merge the member revision into the working file and move the lock (if any) to the member revision, select <b>Resynchronize</b>. The check in operation is not completed. You should perform additional testing on the merged file before checking it in.</li> <li>■ To resynchronize the member revision into the working file by change package, and move the lock (if any) to the member revision, select <b>Resynchronize By CP</b>. The check in operation is not completed. You should perform additional testing on the merged file before checking it in. To use this option, the member revision must be associated with a change package.</li> <li>■ To be asked to confirm the action to be taken, select <b>Confirm</b>.</li> <li>■ To cancel the check in operation, select <b>Cancel</b>.</li> </ul> <p><b>Note:</b> If you select a specific <b>Revision Number</b>, or the <b>Force Creation of New Branch</b> option, no action is necessary when checking in a revision that is not the member revision.</p>

---

# Renaming a Member

---

**CLI EQUIVALENT** `si rename`

---

When you need to change the name of a file in your project, for example, the title of a user guide, you can rename the member while working from a **Project** or **Sandbox** view. Renaming a member performs a drop of the old member name, and an add of the new member name and creates a new revision in the archive, then copies the attributes from the old member name to the new member name.

The member's archive remains the same and points to the old member name so that change packages, member histories, and project histories continue to work.

To rename a member in the GUI, select the member you want to rename, and then select **Member > Rename**.

---

**NOTE**

- Because renaming a member affects other users on the same development path, the member you want to rename must not be locked by any other user.
  - The new name must be in the same directory as the existing old name. To move a member into a new directory, use the Move Members command.
- 

## ***Rename a Member That Is the Tip Revision***

The *tip revision* is the most recent revision on a branch in a history.

When you rename an unlocked member, MKS Integrity places a lock on that member and creates a new, unlocked revision at the tip. For example, if you rename an unlocked member (1.3), MKS Integrity locks revision 1.3 and checks in a duplicate file for revision 1.4 with the new member name, leaving 1.4 unlocked.

If you rename a locked member, MKS Integrity performs the rename operation and retains your lock. For example, if you rename a locked revision (1.3), MKS Integrity performs the rename by checking in a duplicate file for 1.4 and then moves your lock to revision 1.4.

If you rename a member in a variant project while you also have the member locked in the master project, MKS Integrity performs the rename operation but moves your lock to the variant member.

## ***Rename Members on a Branch***

If you rename a member that is not the tip revision, that is, a revision along the branch, MKS Integrity creates a duplicate of that revision along a new branch and updates the member revision to that branch.

## Rename Member Options

Options	Description
<b>Rename Working File</b>	Displays only when the command is performed from the <b>Sandbox</b> view. Renames the working file in your Sandbox and preserves any changes made to that file. If not set, the working file is not renamed and becomes a former member which will be confirmed for deletion the next time the Sandbox is resynchronized.
<b>Overwrite Existing File</b>	Replaces an existing working file in the Sandbox for the new file name.
<b>Confirm Rename</b>	Confirms that you want to rename the selected member.
<b>Defer Rename</b>	Displays only when the command is performed from the <b>Sandbox</b> view. Delays the rename operation in the project until the deferred operation is submitted. The rename operation in the Sandbox still takes place immediately. If change package reviews are mandatory, and this option is not selected, a pending member (and a pending change package entry) is created when the operation is complete.
<b>Create Branch if Variant (GUI only)</b>	Creates a branch for the renamed member.
<b>Close Change Package</b>	Closes the specified change package after performing the rename operation. This option only appears if your administrator has enabled the use of change packages.

---

# Discarding Changes to a Member

---

**CLI EQUIVALENT** `si revert`

---

If you are sure you do not want to check in a member's modified working file, you can revert it to its state before it was checked out. This command is useful if you made changes that you do not want to keep for a member.

Reverting a member discards any changes made to the working file since it was checked out, and then unlocks the revision.

---

**NOTE** If the revision you have locked is not the revision the working file is based on, then the lock is retained.

---

When you revert a deferred add operation, MKS Integrity retains the working file in your Sandbox. After the revert operation, the file becomes a non-member, rather than a former member, because it was never added to the project.

## To discard changes to a member's working file in the GUI

- 1 From a **Sandbox** view, select one or more members that are checked out in your name or are modified.
- 2 Select **Member > Revert**.

If you have made changes to your working file, the **Confirm Overwrite Working File** dialog box displays. If you want to retain your changes in the working file, click **No** (**No to All** for multiple members). If you want to compare your working file with the revision you are reverting it to, click **Differences**. To proceed with the revert operation for a member, click **Yes** (for multiple members, click **Yes to All**).

The member is reverted to its original state prior to checkout.

# Discarding Changes to a Member

If you are sure you do not want to check in a member's modified working file, you can revert it to its state before it was checked out. This is useful if you—accidentally or purposefully—save changes that you do not want to keep.

Reverting a member discards any changes made to a member's working file since it was checked out, and then unlocks it.

---

**NOTE** If the revision you have locked is not the working revision, the lock will be retained.

---

When you revert a deferred add operation, MKS Integrity retains the working file in your Sandbox. After the revert operation, the file becomes a non-member, rather than a former member, because it was never added to the project.

## To discard changes to a member's working file in the GUI

- 1 From a **Sandbox** view, select one or more members that are checked out in your name or are modified.
- 2 Select **Member > Revert**.

If you have made changes to your working file, the **Confirm Overwrite Working File** dialog box displays. If you want to retain your changes in the working file, click **No** (**No to All** for multiple members). If you want to compare your working file with the revision you are reverting it to, click **Differences**. To proceed with the revert operation for a member, click **Yes** (for multiple members, click **Yes to All**).

The member is reverted to its original state prior to checkout.

# Resyncing Members

---

**CLI EQUIVALENT** `si resync`

---

When many users are working from Sandboxes based on the same master project, it is common for the members in an individual Sandbox to become out of sync with the member revisions in the project. For example, the member revision of a particular file may be at 1.5, while you still have revision 1.2 in your Sandbox.

When this happens, a member delta symbol () displays next to the member in the **Sandbox** view, signaling its status.

To update out of sync working files to the most current member revisions, you *resynchronize* the members.

---

**CAUTION** Resynchronizing members can overwrite files, even ones that you have locked and that have changed since you last checked them out. Be certain of your response to any prompts indicating that the files will be overwritten—once replaced, you cannot get back your changes. If files have been dropped from a project, resynchronizing deletes them.

---

If the revision that your working file is based on is locked by you, your lock is automatically moved to the member revision before the resync proceeds. If you had an exclusive lock on the working revision, and another user already has an exclusive lock on the member revision, you are prompted to downgrade your lock to non-exclusive.

If you are working in a sparse Sandbox, resynching deletes your working file, unless the revision that it is based on is locked by you.

When working in your Sandbox, you can also use the Resynchronize By Change Package command. When you select a member and use Resynchronize By Change Package, MKS Integrity automatically searches the change package associated with the member you are resynchronizing and then brings all of the changes recorded in the change package from the project to your Sandbox.

## To resynchronize a member in the GUI

- 1 From a **Sandbox** view, select one or more members that contain member deltas()�.
- 2 Select **Member > Resynchronize**.

If you have made changes to your working file (without a lock), and then attempt to resynchronize the member, the **Confirm Overwrite Working File** dialog box displays. If you want to retain your changes in the working file, click **No** (**No to All** for multiple members). If you want to compare your working file with the revision you are resynchronizing it with, click **Differences**. To resynchronize the member by overwriting the working file, click **Yes** (for multiple members, click **Yes to All**).

You can also merge your working file with the revision you are resynchronizing it with.

The selected member is updated.

# Locking a Member

---

**CLI EQUIVALENT** `si lock`

---

A lock is a feature of MKS Integrity that controls how changes are made to revisions. When a change is checked in, it requires the revision being updated to be locked, in order to prevent more than one user from simultaneously checking in changes to the same revision.

Normally you lock a member during a checkout. Sometimes, however, you may have made changes to a working file that was not checked out in your name first. In this case, you can set a lock without overriding your changes.

To lock a member, select the member or revision in the GUI, then select **Member > Locks > Lock**.

The person who has a revision locked is referred to as the *locker*.

You can upgrade your lock from a non-exclusive lock to an exclusive lock in the Locks view.

The following are the options on the Lock Revision dialog box:

Options	Description
<b>Lock Type</b>	Allows you to determine the type of lock obtained. Select one of the following options: <ul style="list-style-type: none"><li>■ To obtain an exclusive lock on the member, select <b>Exclusive</b>. An exclusive lock prevents other users from checking in changes.</li><li>■ To obtain a non-exclusive lock on the member, select <b>Non Exclusive</b>. A non-exclusive lock enables other users to check in changes.</li><li>■ To obtain a lock type based on the locks policy, select <b>Follow Policy</b>. For information on the locks policy, contact your administrator.</li></ul>
<b>Downgrade Lock Request on Lock Conflict</b>	Obtains a non-exclusive lock when you try to get an exclusive lock on a revision that another user has an exclusive lock on.
<b>Revision Mismatch is Error</b>	Causes an error to display if the working revision does not match the member revision. This option only displays if you are performing the Lock operation from a Sandbox.

# Viewing Member Locks

---

## CLI EQUIVALENT `si locks`

---

You can view a list of all of the members you have locked across all projects, even projects that you no longer have permissions to access. You can then remove unused locks, downgrade exclusive locks to non-exclusive locks, or upgrade non-exclusive lock to exclusive locks.

---

**NOTE** When viewing locks in the GUI, the locks displayed are based on the username you using. However, the username used to determine which locks are displayed is case sensitive. Ensure that you use the same text case when logging into the server to ensure that all locks are displayed.

---

You can also view a list of all the members that another user has locked, as long as they are in projects that you have permission to view. This is useful if you want to know what files a specific user is working on. Depending on the permissions defined by your administrator, you may be able to downgrade or remove locks for another user.

Operation	Procedure
To view member locks in the GUI	Select <b>Member &gt; Locks &gt; View</b> . The <b>Locks</b> view displays.
To view member locks in the Web interface	Select <b>Tools &gt; Manage My Locks</b> . The <b>Locks</b> view displays.
To upgrade a non-exclusive lock to an exclusive lock in the GUI	Select an item in the <b>Locks</b> view, and then select <b>Member &gt; Locks &gt; Upgrade My Lock</b> .
To upgrade a non-exclusive lock to an exclusive lock in the Web	Select an item in the <b>Locks</b> view, and then select <b>Locks &gt; Upgrade My Lock</b> .
To remove another user's lock in the GUI	Select an item in the <b>Locks</b> view, and then select <b>Member &gt; Locks &gt; Remove Lock</b> .
To remove another user's lock in the Web interface	Select an item in the <b>Locks</b> view, and then select <b>Locks &gt; Remove Lock</b> .
To change the user whose locks are displayed in the Locks view (GUI only)	From the <b>Locks</b> view, select <b>View &gt; Change User</b> .
To find another user's locks (GUI only)	Select <b>Member &gt; Find &gt; Locks</b> .

---

## NOTE

You may only view other users' locks that are in projects you have permission to view. You can always view your locks for any project regardless of project permissions.

You cannot remove another user's lock on a member that was locked from a shared subproject.

---

## Locks View

The **Locks** view displays the following information by default:

Column	Description
<b>Project</b>	Name and path of the project where the member revision was locked from. If the member revision was locked from a shared subproject, it is the subproject name and path that are displayed.
<b>Member Name</b>	Displays the member name for the locked revision.

---

Column	Description
<b>Lock Type</b>	Indicates whether the lock is exclusive or non-exclusive.
<b>Revision</b>	Displays the locked revision number.
<b>Time</b>	Displays the time the revision was locked

---

# Removing or Downgrading Locks on Members

---

**CLI EQUIVALENT** `si unlock`, `si unlockarchive`

---

When you no longer need the exclusive ability to change a member, you can unlock it. The main reason to unlock a member is to abandon work on the member, in which case you can use the Revert command. In views where there is no Sandbox context for the member you are unlocking, you can use the Remove My Lock command.

You can downgrade your exclusive lock to a non-exclusive lock, in order to enable another user to get the exclusive lock. Depending on the permissions your administrator has defined, you may also be able to downgrade another user's exclusive lock to a non-exclusive lock, in order to be able to give yourself an exclusive lock. To remove locks for another user, use the Locks view. For information on permissions and on your locks policy, contact your administrator.

Operation	Procedure
To unlock a member in a Sandbox in the GUI	Select <b>Member &gt; Revert</b>
To unlock a member in the GUI interface when there is no Sandbox context	Select <b>Member &gt; Locks &gt; Remove My Lock</b>
To unlock a member in the Web interface	Select <b>Member &gt; Remove My Lock</b>
To downgrade an exclusive lock on a member in the GUI <sup>a</sup>	Select <b>Member &gt; Downgrade Lock</b>
To downgrade an exclusive lock on a member in the GUI when there is no Sandbox context <sup>a</sup>	Select <b>Member &gt; Locks &gt; Downgrade</b>
To downgrade an exclusive lock on a member in the Web interface <sup>a</sup>	Select <b>Member &gt; Downgrade Exclusive Lock</b>

<sup>a</sup> In the GUI, you can also select a project or subproject to downgrade locks for members in that project recursively.

---

# Making a Member's Working File Writable

---

**CLI EQUIVALENT** `si makewritable`

---

If you want to edit a file locked by someone else, and you have no intention of checking the files back in, you can make the working file writable.

In the GUI, you can make your working file writable during the checkout process or from a **Sandbox** view.

Select a member or revision in the **Sandbox** view and select **Member > Make Working File Writable**.

In the Web interface, you can make your working file writable during the checkout process.

---

# Using Keywords in Working Files

A *keyword* is a placeholder that can be inserted into text-based working files. This placeholder is a special variable (for example, \$Date\$, \$Author\$, \$State\$) used to represent textual information in a working file. Keywords can be expanded (that is, replaced with their literal values) when a revision is checked out.

To use a keyword, simply include it in a working file, surrounded by dollar signs (for example, \$Date\$) and check the file back into its archive.

For example, Chad wants to see information about the member he is editing as a comment in the file. Chad can do this through keywords. He decides to use the \$Log\$ keyword. Chad sets his preferences to enable keyword expansion on checking out a member. He then places the keyword in a member.

---

**NOTE** Your administrator may define custom keywords for your use. For information about these keywords, see your administrator or the online help for the MKS Integrity administration client.

---

The next time you check out the revision, MKS Integrity scans it for keywords and replaces them with the appropriate information, if keyword expansion is turned on.

*Keyword expansion* is the process of automatically adding or updating information to a keyword reference when a revision is checked out or viewed.

For example, if the \$Date\$ keyword is encountered, the date and time of the revision (assigned at check in) is added to the working file as part of the keyword. When expanded, the entry would look something like

```
$Date: 2009/06/12 10:25:32$
```

This method of adding or updating information in a keyword is called *keyword expansion*.

For example, if the member `main.c` has the keywords \$Author\$ and \$State\$ embedded within it, checking out `main.c` and issuing the command:

```
ident main.c
returns
main.c:
$author: paula_t $
$state: Exp $
```

The following MKS Integrity commands contain keyword expansion options:

- Add Members
- Check Out
- Check In
- Resynchronize
- Revert Member

Keyword expansion is configured using the **Preferences** dialog box in the GUI. The dialog boxes in the GUI can override the default settings.

---

**NOTE** Keyword expansion applies to text files only. It is disabled for binary files.

---

---

Text before and after the keyword is preserved, making it suitable for use within expressions, as above, and within comments.

If keyword expansion is enabled and you are checking out a text file that contains the string

```
$Revision$
```

MKS Integrity, when it encounters this string, automatically adds the value of the keyword \$Revision\$ in the format

```
$Revision: value $
```

where

*value* is the appropriate value of the keyword (in this case, the revision number).

For example, including the statement

```
char revnum[] = "$Revision$";
```

in a C source file creates a character string named `revnum` containing the file's revision number. The program can then be configured to display this string when it starts up, automatically presenting the current revision of the program's source file.

Using the \$Revision\$ keyword to obtain the revision number of a file is one of the common applications of keywords. Other common applications include:

- The \$Header\$ keyword provides a one-line summary of useful information associated with a revision. Including this information in a comment makes the information available to anyone looking at the member.
- The \$Log\$ keyword supplies the same sort of information as \$Header\$ plus the revision description. The \$Log\$ keyword provides a complete list of changes that are made to the member over time.

---

**NOTE** The keyword format of \$<keyword>\$ causes MKS Integrity to replace between the first \$ and the next \$. If you use a keyword in the format \$<keyword>, MKS Integrity continues to replace until it encounters another \$. It is possible that MKS Integrity may not encounter another \$ until the file is checked out again. This type of keyword use returns results that are similar to logging.

---

### **Turning Off Keyword Expansion**

If you want keywords to remain exactly as they are in a revision, set your preferences to turn off keyword expansion.

#### **To turn off keyword expansion from the GUI**

- 1 From the **Preferences Configuration** dialog box, select **Configuration Management > Commands**.
- 2 For the **Check Out**, **Resynchronize**, and **Revert** commands, set the **Keywords** option to **Do Not Expand**.
- 3 For the **Check In** and **Add Member** commands, disable the **Unexpand Keywords** option.
- 4 To save the preferences, click **OK**.

#### **To turn off keyword expansion from the CLI**

From the command line, type the following to set and save keyword preferences:

```
si setprefs --save --command=add keywordExpand=noexpand  
si setprefs --save --command=annotate keywordExpand=noexpand
```

---

```

si setprefs --save --command=ci keywordExpand=noexpand
si setprefs --save --command=co keywordExpand=noexpand
si setprefs --save --command=resync keywordExpand=noexpand
si setprefs --save --command=revert keywordExpand=noexpand
si setprefs --save --command=viewrevision keywordExpand=noexpand
si setprefs --save --command=add unexpandKeywords=false
si setprefs --save --command=ci unexpandKeywords=false

```

Setting these options keeps keywords unchanged on all commands. This is useful when importing files from other SCM tools.

### **Locating Keywords**

You can use the **ident** command in the command line interface to locate and display keywords (expanded or unexpanded) in one or more members. For more information about the **ident** command, see the *MKS Integrity 2009 CLI Reference Guide for Configuration Management* or the online man pages.

This command displays the name of each member that contains keywords, as well as the keywords themselves. This provides an easy way to extract identification information from source files, as well as compiled object files.

## **Table of Keywords**

MKS Integrity maintains several keywords that can be used in working files. Keywords are case-sensitive.

Your administrator may create custom keywords. For information on using custom keywords contact your administrator.

The following table describes default keywords and what each expands to.

Keyword	Expands To
\$Author\$	The name of the user who checked in the revision.
\$CompanyInfo\$	The name of the company and/or other company info including address, E-mail and phone numbers. You must enter a keyword policy for the company information that you want displayed. Strings may contain standard escapes like “\n” for new lines, but must be in ISO-646. For information on setting keyword policies, see the online help for the MKS Integrity administration client.
\$Date\$	The checkin date and time of the revision (as assigned at checkin). The time is shown in Greenwich Mean Time (GMT/ or Coordinated Universal Time).
\$Header\$	The file name of the archive, as well as the revision number, date and time, author, state, and the user who has an exclusive lock (if any).
\$Id\$	The same as \$Header\$, except that only the file name of the archive is displayed, not the full path.
\$Locker\$	The user ID of the user who has an exclusive lock on the revision, and the date and time that it was locked (empty if no exclusive lock).

Keyword	Expands To
\$Log\$	The revision description supplied during checkin, preceded by the archive's member name, revision number, author, and revision date. Repeated checkout operations append revision descriptions, rather than replacing existing ones. <b>Note:</b> this keyword does not unexpand.
\$Name\$	The revision label (or labels) attached to a revision.
\$ProjectLabel\$	The label (or labels) associated with the checkpoint of the build project the Sandbox is based on. The \$ProjectLabel\$ keyword can only be used when the Sandbox corresponds to a build project that has a label. If there are multiple labels associated with the selected build project checkpoint, the \$ProjectLabel\$ keyword uses a space-separated list of labels in alphabetical order.
\$ProjectName\$	The fully qualified name of the project of which the archive is a member.
\$ProjectRevision\$	The revision number of the project that the archive is related. For use in build Sandboxes only.
\$ProjectSetting attribute\$	The current value of the attribute defined in MKS Integrity. For example, if the project contains set OS=unix, the keyword \$Setting OS\$ is expanded to \$Setting OS: unix\$
\$RCSfile\$	The archive's unqualified member name.
\$Revision\$	The revision number.
\$SandboxSetting attribute\$	The current value of the attribute defined in MKS Integrity. For example, if the Sandbox contains set OS=unix, the keyword \$Setting OS\$ is expanded to \$Setting OS: unix\$
\$Setting attribute\$	The current value of the attribute defined in MKS Integrity. For example, if the member contains set OS:nt, the keyword \$Setting OS\$ is expanded to \$Setting: nt\$.
\$Source\$	The same as \$RCSfile\$. The archive's unqualified name.
\$State\$	The state setting of the revision. States are defined by your administrator.

# Adding Labels to Members

---

**CLI EQUIVALENT** `si addlabel`

---

A *revision label* is a textual name that describes and refers to a revision. When a member is checked in, you are given the option of assigning it a revision label. Labels can be based on the product release the revision was included in, on the content of the revision, on changes made to the revision, or any other sort of information that would be useful in identifying that particular revision. Revisions in a history can be displayed and selected either by revision number or revision label.

Labels cannot contain colons (:), square brackets ([ ]), leading spaces, numbers in the same format as a valid revision number (1.23), or numbers without any spaces (14325). Numbers that contain spaces are acceptable, for example, 2432 1234.

Although you generally add a label to a new revision upon check in, there may be times when you want to add an additional label or change the label assigned to a revision. For example, you might want to label each member in a project when you checkpoint the project.

---

**NOTE** In the Web interface, MKS Integrity displays up to three member labels in the **Labels** column of the **Project** view. If a member has more than three labels, MKS Integrity displays a link ([...](#)) that you can click to view all the member labels. Labels appear in alphabetical order in selection lists.

---

Interface	Procedure
GUI	Select <b>Member &gt; Properties &gt; Add Label</b>
Web	Select <b>Member &gt; Add Label</b>

---

If the label you are adding already exists on another revision, click **Options**, and select the **Move Existing Label** option.

---

**TIP** You can also add labels to the member in the **Member Information** dialog box or **Revision Information** dialog box.

---

---

# Deleting Member Labels

---

**CLI EQUIVALENT** `si deletelabel`

---

Sometimes you may want to delete a member label. For instance, you may decide the label no longer accurately reflects that particular revision. In addition, if you have assigned the same label to a number of members, you might want to remove them all with one command.

Interface	Procedure
GUI	Select <b>Member &gt; Properties &gt; Delete Label</b> . In the <b>Delete Label</b> dialog box, select a label to delete from the <b>Label</b> list.
Web	Select <b>Member &gt; Delete Label</b> . In the <b>Delete Label</b> dialog box, select a label to delete from the <b>Label</b> list.

---

**TIP** You can also delete a member's labels in the **Member Information** or **Revision Information** dialog box.

---

# Freezing Members

---

**CLI EQUIVALENT** `si freeze`

---

When your development team has largely finished a portion of a project and some project members are in a stable state, you can freeze individual members within a project or Sandbox.

Freezing a member places it in a state that prevents changes from being made to the member information that resides in the project file. For example, you cannot update the member revision or change the attributes of a frozen member. Freezing is the opposite of thawing a member.

Freezing restricts member information from being updated, preventing these members from being changed by accident. However, development work can still continue in the member file itself. For example, if new revisions are checked into the member's archive, MKS Integrity does not update the member revision for the project.

Freezing is useful for facilitating:

- project checkpointing
- member promotion
- software distribution

Freezing prevents changes to member information in the project, but does not affect the member file itself. Revisions can still be checked out, modified, and checked in, but none of the changes are included as part of the member information in the project.

You can change the label or state of frozen members, but not their attributes. Freezing can be used immediately before a checkpoint operation to ensure no one changes the project or its members before the checkpoint is complete.

When you want to allow project members to be changed, you can thaw them.

If a member is frozen, MKS Integrity reports the availability of new revisions when anyone checks them into the archive. MKS Integrity does not update the project to the latest revision, so an appropriate person must make the decision to thaw the member and update the project as a whole.

Interface	Procedure
GUI	Select <b>Member &gt; Properties &gt; Freeze</b> .
Web	Select <b>Member &gt; Freeze</b> .

## Example

A sample freezing sequence is as follows:

- Working with the `Apex.pj` project, a release engineer freezes project member `utility.dll` at revision 1.2.
- The snowflake symbol displays beside `utility.dll`, revision 1.2 in the `Apex.pj` **Project** view. (The snowflake symbol displays only in the context of the project.)
- A developer checks out `utility.dll`, revision 1.2, modifies it, and checks it back in.
- The new version of `utility.dll` is not accessible to the `Apex.pj` project until revision 1.2 is thawed.

# Thawing Members

---

**CLI EQUIVALENT** si thaw

---

When you decide to allow project members to evolve again, you can thaw any frozen ones.

Thawing a member removes the restriction on changing member information in the project and makes previously checked in member information available to the project. Thawing a member is the opposite of freezing a member.

Interface	Procedure
GUI	Select <b>Member &gt; Properties &gt; Thaw</b> .
Web	Select <b>Member &gt; Thaw</b> .

## **Example**

A sample thawing sequence is as follows:

- As part of the development cycle of the `Apex.pj` project, the release engineer freezes the file `utility.dll`, revision 1.2.
- A developer checks out `utility.dll`, revision 1.2, modifies it, and checks it back in.  
The new version of `utility.dll` (that is, version 1.3) is not accessible to the `Apex.pj` project until revision 1.2 is thawed.
- The release engineer thaws `utility.dll`, revision 1.2. Thawing revision 1.2 of `utility.dll` makes the changes available to the `Apex.pj` project. MKS Integrity notifies that a newer revision of `utility.dll` (version 1.3) is available to the project.
- Once the developer's modifications are reviewed and accepted, the release engineer incorporates the modifications by choosing **Member > Properties > Update Revision** in the GUI or **Member > Update Revision** in the Web interface. MKS Integrity updates `Apex.pj` to include the modifications previously checked in by the developer.
- Revision 1.3 becomes the head revision for the project member `utility.dll`.

# Submitting Deferred Operations

---

**CLI EQUIVALENT** `si submit`

---

To submit, or complete, a deferred operation, you use the **Member > Submit Deferred** command. Submitting the operation completes the command and makes it visible in the associated project.

For example, Chad completed his unit testing of the savings calculator with the new image. He can now submit his deferred drop and add of the image files.

---

**IMPORTANT** If exclusive locking is enabled, you cannot submit a member with a deferred check in operation if you do not have a lock on the member. You must check out or lock the member before submitting the deferred check in operation. For more information on exclusive locking, see the *MKS Integrity Server 2009 Installation and Configuration Guide*.

---

By default, the change package that was specified during the deferred operation is used for the Submit Deferred command. To select a different change package or create a change package, click **Options** and enable **Override the change package to a specified value**.

If change package reviews are mandatory, you do not need to submit deferred operations prior to submitting the change package.

The following are the options on the Submit Deferred dialog box.

Options	Description
<b>Use the change package provided when the element was deferred</b>	Selects the change package that was originally associated with the deferred item.
<b>Override the change package to a specified value</b>	Provides the means to select (or create) a different change package from the one that was originally associated with the deferred item.
<b>Close Change Package</b>	Closes the change package after submitting the item and completing the associated deferred operation. This option only appears if your administrator has enabled the use of change packages.

---

# Resyncing Members With Deferred Operations

When you resynchronize a member associated with a deferred operation, MKS Integrity performs the following deferred operations:

- **Deferred Add**  
When resynchronizing a member associated with a deferred add, MKS Integrity does not make any changes to the working file.
- **Deferred Drop**  
When resynchronizing a member associated with a deferred drop, where the working file is modified, MKS Integrity asks you if you want to overwrite the existing working file.
- **Deferred Rename**  
When resynchronizing a member associated with a deferred rename, MKS Integrity asks you if you want to overwrite the existing working file. If the new working file is missing, MKS Integrity resynchronizes it. The working file with the old name is then deleted from the Sandbox.
- **Deferred Move**  
When resynchronizing a member associated with a deferred move, MKS Integrity informs you that there is a deferred operation on the member and asks you if you want to proceed. MKS Integrity then resynchronizes the working file to the version in the project and in the new location. The deferred indicator remains after the resync completes.
- **Deferred Checkin**  
When resynchronizing a member associated with a deferred checkin, MKS Integrity asks you if you want to overwrite the existing working file to correspond to the member revision.
- **Deferred Update Revision**  
When resynchronizing a member associated with a deferred update revision, MKS Integrity asks you if you want to overwrite the existing working file.
- **Deferred Import**  
When resynchronizing a member associated with a deferred import, MKS Integrity asks you if you want to delete the existing working file.

---

## Cancelling Deferred Operations

Once you defer an operation, you must use the **Member > Revert** command to cancel the defer operation.

When you revert a deferred add operation, MKS Integrity retains the working file for the deferred member. After the revert operation, the file becomes a non-member, rather than a *former member*, because it was never added to the project.

# Viewing a Member History

CLI EQUIVALENT `si viewhistory`

MKS Integrity maintains detailed information about each revision in the member history. In the GUI, revision information displays in the **Member History** view.

Interface	Procedure
GUI	Select <b>Member &gt; Views &gt; View History</b>
Web	Select <b>Member &gt; View Member History</b>

You can also view and modify revision information through the **Revision Information** view.

MKS Integrity maintains an archive of all the changes made to a member since it was put under revision control.

An *archive* is a file containing the history of a member (a record of all the changes made to it since it was put under revision control). From the information contained in the history, MKS Integrity can reconstruct any previous version of the member.

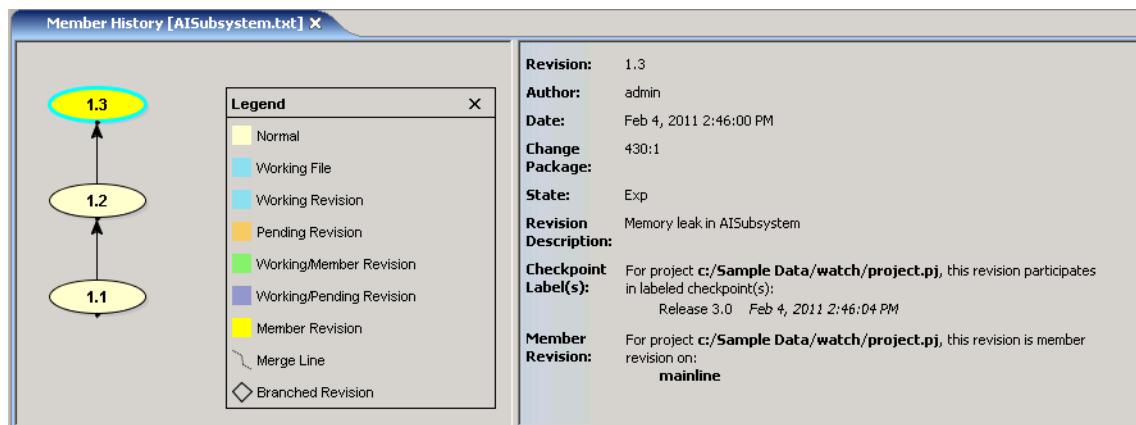
MKS Integrity maintains historical information called *archive information*. This information includes revision labels, users who have locks on revisions in the archive, the starting point of the default branch revision, the data type (text or binary), whether the archive is compressed, whether exclusive locking applies to the archive, and a description of the archive.

You can view and modify archive information through the **Archive Information** view.

## Working With the Member History View

By default, the member history view displays two panels: a panel showing the path of development, and a panel showing details for a selected revision. In order for information to display in the details panel, you must select a checkpoint in the GUI or click a checkpoint number in the **Revision** column in the Web interface.

In the GUI, you can toggle between a graphical view and a list view by clicking **View > List** or **View > Graphical**.

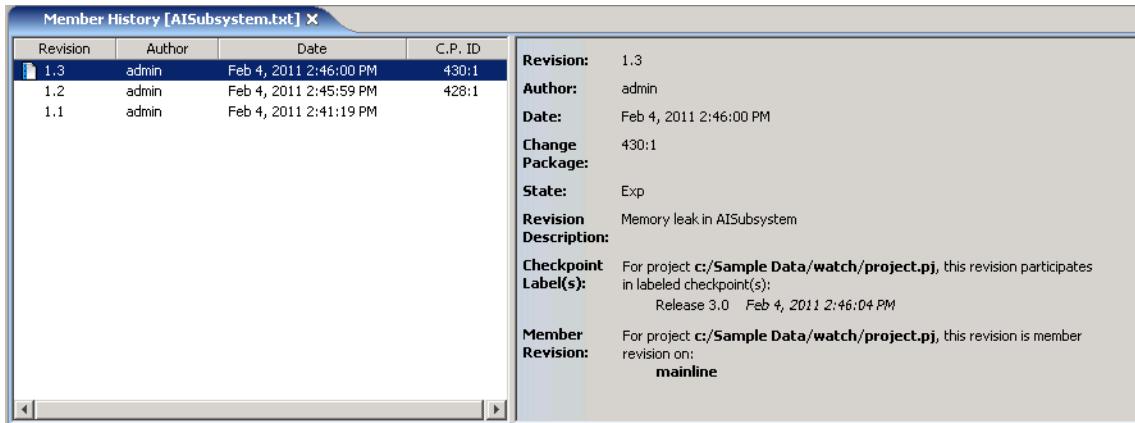


Graphical Member History view

The information in the graphical view shows the path of development from revision to revision, including branches and merge lines. You can change the information that displays beside each

graphical revision by selecting **View > Show beside each node**. If summary information is not displayed in the view, it displays in a tooltip when you place your mouse pointer on a specific revision. The available selections are

Selection	Description
<b>Nothing</b>	Nodes unadorned
<b>Revision Label(s)</b>	Member revision labels displayed beside node
<b>Checkpoint Label(s)</b>	Labels of checkpoints in which the member revision participated displayed beside node
<b>Summary</b>	Summarizes author and date of revision.



*List Member History view*

The information in the list view is displayed in columns. For more information on the default columns, see “Member History View Default Columns” on page 362.

The details panel in the graphical or list view displays additional details for a selected revision. You can turn the display of this panel on or off using the **View > Show Details** option.

**TIP** In the GUI, some fields may contain lists too long to display by default. Click **More** to display the complete list.

In the Web interface, the content of all lists are displayed and may require scrolling to see the complete list of fields and their contents.

The following information can display in the member history details.

Field	Description
<b>Revision</b>	Revision number of the member.
<b>Author</b>	User who checked in the revision.
<b>Date</b>	Date the revision was created.
<b>Change Package</b>	Change package associated with the revision.
<b>State</b>	State of the revision. States are defined by your administrator.
<b>Revision Description</b>	Brief description of the revision.

Field	Description
<b>Locked By</b>	All locks on the selected revision, including information for each lock. Each lock is listed separately, with the exclusive lock (if any) listed first, followed by all other locks, sorted by timestamp (newest first). <b>Note:</b> The list of locks includes any that exist outside the project and development path context of the selected revision. <b>TIP:</b> Click <b>More</b> to see additional Sandbox information.
<b>Checkpoint Label(s)</b>	Labels of checkpoints in which the selected revision participated, including date and time information.
<b>Member Revision</b>	The project and development path that the revision is the member revision in. <b>Note:</b> If a revision is member revision in a shared subproject, only the original project path is displayed. Any projects where the subproject was added as shared are not displayed. Only projects that you have the <code>OpenProject</code> permission for display. <b>Note:</b> Member revision information is not supported for an RCS type repository. The maximum number of projects displayed is controlled by the server limit, which is set at 5000 by default. Administrators can change the default limit. If the number of projects exceeds the set limit, MKS Integrity displays a message indicating that there are too many results to display in the view.

### Filtering the Member History View

You can use the revision filter in the GUI to display any revisions of the selected member that meet specified selection criteria. You can change the filter by selecting **View > Change Filter**.

## Member History View Default Columns

**Note** The **Checkpoint Label(s)** column is not displayed in the GUI by default. The column displays labels of checkpoints in which the selected revision participated. However, only the first three revisions are displayed in the field.

To see the **Checkpoint Label(s)** column in the Web interface, your administrator must first configure the server. If you require the column, contact your administrator.

Column	Description
<member path and name>	Member's path and name, for example, <code>c:/sandboxdemo/demoap.c</code> . This only displays in the GUI when the member history is viewed from a Sandbox. It displays in the first row underneath the columns.
<b>Revision</b>	<b>Revision</b> displays the following information: <ul style="list-style-type: none"> <li>■ revision that the working file corresponds to, indicated by a white document icon</li> <li>■ member's revision, indicated by a blue striped document icon</li> <li>■ list of the revisions in the member history, for example, 1.1, 1.2, 1.3, and so on</li> </ul>
<b>Author</b>	User who checked in the revision.

Column	Description
<b>Date</b>	Date and timestamp of the revision. Date and timestamp can be either the checkin time and date or the date and time the file was last modified, depending on how the user checks in the file.
<b>Locked</b>	Users who have a lock on the revision. The padlock icon indicates the type of lock and any potential conflicts with other lockers. If there are multiple lockers, your lock is listed first, followed by the locker who has an exclusive lock (if any), followed by any lockers with non-exclusive locks. If there are multiple lockers, click on the field to display a drop-down list of lockers.
<b>Revision Labels</b>	Labels attached to the revision, for example, Draft1. This displays in the GUI only.
<b>State</b>	Revision's state, for example, Beta. This displays in the GUI only. States are defined by your administrator.
<b>C.P. ID</b>	Displays the revision's associated change package ID, for example, 8230:3. MKS Integrity displays a lock change package ID (a change package assigned during a checkout) in priority over the member revision change package ID (the change package assigned during a checkin).
<b>Revision Description</b>	Comments that were assigned to the revision when it was checked in. This displays in the GUI only.

# Filtering a Member or Project History in the GUI

You can use the revision filter in the GUI to display any revisions of the selected member or project that meet specified selection criteria. You can change the filter by selecting **View > Change Filter**. This allows you to display a specific group of revisions for a member or project that might have hundreds or thousands of revisions. Revisions selected with this command can be manipulated as a group by other MKS Integrity commands. You can apply filters to a member or project history in either the graphical history view or the list view.

Even with filtering, the number of revisions displayed can be too large to work with. By default, a maximum of 200 trunk revisions are displayed (the revisions branched from these trunk revisions are also displayed). You can change the maximum by selecting **View > Maximum trunk revisions**. The number of revisions displayed is calculated based on the number of visible mainline revisions; branched revisions are not included in the count.

For example, specifying the range of branched revisions 1.14.1.1 to 1.14.1.999 might return a results screen with 4000 branched revisions of the form 1.14.x.x.x.x. The 200 trunk revision display maximum is not reached because all of the branched revisions are for the single trunk revision 1.14.

**NOTE** If you limit the number of revisions to display and the maximum has been reached, a banner displays at the top of the view indicating the number of revisions displayed.

## Selecting Filter Criteria

You choose the selection criteria you want to use by selecting one or more check boxes in the **Filter Revisions** dialog box, specifying filter values where required.

To invert a filter, click the filter a second time and the ! symbol displays. You would invert a filter if you wanted to display, for example, revisions that are not locked.

If you choose more than one selection criteria, select whether the criteria should be combined using a **Logical AND** or a **Logical OR**.

Filter	Description
<b>All Revisions</b>	All revisions of the selected member (no filter). You cannot use the <b>All Revisions</b> filter in combination with other filters. Selecting the <b>All Revisions</b> filter deactivates other filters.
<b>In Development path</b>	Project checkpoints that are on the specified development path. This option does not display for member histories.
<b>On Branch</b>	Revisions that are on the specified branch.
<b>In Range</b>	Revisions that are in the specified revision number range. If required, you can enter just one end of the range. The range includes the low and high values. Check the <b>Include Subbranches</b> option to include any branched revisions within the range.
<b>Created during the past</b>	Revisions that were created during the specified amount of time in the past.
<b>Created in Time Range</b>	Revisions that were created within the specified time range. You can also specify one end of the range.
<b>Labeled revisions</b>	Revisions with labels. If a label is specified, only revisions with that label are displayed. If no label is specified, all labeled revisions are displayed.
<b>With Label like</b>	Revisions with labels similar to the specified label. Recently used terms can be selected from the drop-down list. Wildcards such as * and ? are supported.

---

Filter	Description
<b>Locked By</b>	Revisions that are locked. If a user is specified, only revisions locked by that user are displayed. If no user is specified, all locked revisions are displayed. This option does not display for project histories.
<b>Lock Type</b>	Revisions locked with the specified type of lock.
<b>At State</b>	Revisions at the specified state. States are defined by your administrator.
<b>By Author</b>	Revisions created by the specified author. You can choose yourself by selecting your name or <b>-Me-</b> from the user list.
<b>Pending Revisions</b>	Revisions that are pending.
<b>Special Revisions</b>	Member and working revisions.
<b>Branch Boundaries</b>	Revisions that are branched, or at the base or tip of a branch.

# Viewing Member Archive Information

**CLI EQUIVALENT** `si archiveinfo`

A member's content is stored in an archive. You can view information about the member archive and set locking, compression and storage properties for the archive.

In the GUI, select **Member > Archive > View Information**, or in the Web, select **History > Archive Information**. The **Annotated Revision** view displays.

By default, the **Archive Information** view displays the following information:

Tab	Options
<b>General</b>	<p><b>Member Name</b> is the path and name of the member that the archive is for. <b>Project/Sandbox Name</b> is the path and name of the member's project or Sandbox.</p> <p><b>Archive Name</b> is the path and name of the displayed archive.</p> <p><b>Archive Type</b> displays the type of data stored in the archive.</p> <p><b>Default Branch</b> specifies the starting point of the default branch. To specify a default branch, enter a branch number in the <b>Default Branch</b> field, for example, 2.1.1.</p> <p><b>Exclusive Lock Mandatory</b> specifies if an exclusive locks policy is in effect for the archive. With an exclusive locks policy, you must have an exclusive lock before checking in any changes.</p> <p><b>Compressed</b> specifies if the archive is compressed. To compress the archive, select the <b>Compressed</b> option. Compression is only supported on file system type repositories.</p> <p><b>Store by Reference</b> causes each revision to be saved to a separate file, instead of saving all revisions to one file. This feature improves performance for archives that contain large binary files, but can also be used for text files. To store the archive by reference, select the <b>Store by Reference</b> option. This feature is available for archives stored in either a file system repository or a database repository.</p> <p><b>Shared</b> specifies if other members share the archive. This option displays only if you are using the database repository.</p> <p><b>Archive Description</b> describes the archive. If necessary, enter or edit a description.</p>
<b>Labels</b>	Revision labels in the archive, for example, Draft1 1.1.
<b>Locks</b>	For each lock on the archive, it displays the following information: <ul style="list-style-type: none"><li>■ <b>Locker</b> displays the name of the users who locked the revision.</li><li>■ <b>Lock Type</b> displays whether the user has an exclusive or non-exclusive lock on the revision.</li><li>■ <b>Revision</b> displays the revision number of the locked revision.</li></ul>

---

# Setting the Default Member Revision

---

**CLI EQUIVALENT** `si updaterevision`

---

You can update a member revision without checking in a member by using the **Update Revision** command.

The member revision is the default revision that users work with in all other Sandboxes. For example, if `demoapp.c 1.1` is the member revision, setting `1.2` as the member revision makes it the default revision in all Sandboxes.

For example, code freeze has been reached on the stock calculator, and Steve has frozen all of the members in that project. When all the project members are frozen, Quality Assurance tests the code. In this case, they have found a bug in the stock calculator. Steve must now incorporate the fix into the project by thawing the member revision, updating it to the head revision, and freezing it again.

You can update the member revision to a pre-selected or pre-defined revision. This is useful when you want the member revisions in a project to reflect revisions based on a symbolic location in the development path (working file, head revision, trunk tip, or member branch tip) or property (state, label, or timestamp).

Interface	Procedure
GUI	Select <b>Member &gt; Update Member Revision</b> .
Web	Select <b>Member &gt; Update Revision</b> .

---

**NOTE** You cannot update a frozen member revision. You must first thaw the member revision and then update it.

---

## Set Member Revision/Rule Dialog Box (GUI)

Tab	Description
<b>Selection</b>	<p>Set a pre-defined or specific revision as the member revision.</p> <p><b>Pre-Defined Revision</b> option is a symbolic revision that represents a location in the development path. To select a pre-defined revision to update the member revision to, click <b>Pre-Defined Revision</b>, then choose a pre-defined revision from the list. Available pre-defined revisions are:</p> <ul style="list-style-type: none"> <li>■ <b>Working</b> updates the member revision to the version of the working file.</li> <li>■ <b>Head</b> updates the member revision to the member's head revision.</li> <li>■ <b>Trunk Tip</b> updates the member revision to the latest revision in the trunk, independent of the default branch settings.</li> <li>■ <b>Member Branch Tip</b> updates the member revision to the latest revision along the member's current branch of development.</li> <li>■ <b>Member Rule</b> updates the member revision to the revision defined as the member rule. The member rule is configured using the <b>Set Member Rule</b> command.</li> </ul> <p>The <b>Specific Revision</b> option allows you to select a revision number, or a revision based on a property (state, label, or timestamp), current project configuration, or external project configuration. To select a specific revision, click <b>Specific Revision</b>, then click one of the following tabs: <b>Revisions</b>, <b>Properties</b>, <b>Project</b>, or <b>Link</b>.</p>
<b>Revisions</b>	Select a specific revision as the member revision.
<b>Properties</b>	<p>Select a revision with a specific state, label, or timestamp as the member revision.</p> <p><b>Last revision with state</b> allows you to make a revision with a specific state the member revision.</p> <p><b>Note:</b> This option displays only if promotion is enabled by your administrator. The options in the list depend on the states configured by your administrator. For more information on the available states, see your administrator.</p> <p><b>Revision with label</b> allows you to make a specific labeled revision the member revision.</p> <p><b>Note:</b> This option displays only if labels exist in the member history.</p> <p><b>Last revision at date</b> allows you to make the most recent revision at the specified timestamp the member revision. Because the revision may be on a different branch closest to the specified timestamp, you can limit the matching of the timestamp to the revision to a <b>Member Branch</b>, <b>Any Branch</b>, or a specific branch in the list.</p>
<b>Link</b>	<p>Set the member revision to whatever is the member revision for the corresponding member in a specific external project configuration (normal, variant, build). To link the project that the member belongs to (the target project) with the master project, configure the following options:</p> <p>To specify the master project, click <b>Member Revision in a Specified Project</b>. Next, click <b>Select</b> to browse for the specific project and configuration type.</p> <p>Under <b>Member</b>, choose an option to specify the member in the target project:</p> <ul style="list-style-type: none"> <li>■ <b>Find member in specified project</b> searches for the member with the same backing archive in the target project.</li> <li>■ <b>Find member recursively</b> searches recursively throughout the subprojects for the member. There must be exactly one backing archive for each member.</li> <li>■ <b>Specify member name</b> is the name of the member in the target project. This is useful when the target project contains several members that point to the same archive as the current member.</li> </ul>

## Set Member Revision/Rule Options (GUI)

Option	Description
<b>Defer Update Revision</b>	Commits the operation later. This option displays for the <b>Set Member Revision</b> dialog box only. If the change package reviews are mandatory, select this option to create a pending entry for this operation at the time of change package submission. If this option is not enabled, MKS Integrity creates the pending entry at the completion of this procedure..
<b>Close Change Package</b>	Closes the change package when the operation is complete. This option displays for the <b>Set Member Revision</b> dialog box only.
<b>Store Expanded Revision</b>	Sets the rule to the numeric revision number instead of the symbolic revision. For example, if you set the rule according to the <code>ReadyForQA</code> label and select the <b>Store Expanded Revision</b> option, the revision that currently corresponds to the <code>ReadyForQA</code> label is stored and always used when the rule is applied, even if the label later moves to another revision. This option displays for the <b>Set Member Rule</b> dialog box only.
<b>Confirm Rule Clearing</b>	Causes MKS Integrity to confirm that the rule will be cleared. This option displays for the <b>Set Member Rule</b> dialog box only.
<b>Override Rule</b>	Overrides the existing member rule. This option displays for the <b>Set Member Rule</b> dialog box only.

# Setting a Revision Rule for Members

A member rule is a revision—typically a symbolic revision—attached to a member. You can use it with any command that allows you to specify a revision based on the member rule.

In the GUI, only the **Check Out** and **Update Member Revision** commands allow you to specify a revision based on the rule defined for the member.

You can set a member rule to specify a revision based on a symbolic location in the development path (working file, head revision, trunk tip, or member branch tip), property (state, label, or timestamp), current project configuration, or external project configuration.

For example, if you wanted to share members in project A with project B without using shared subprojects, you could apply a member rule to members in project B that links them to the corresponding members in project A. Then, you would update the members in project B by using the **Update Member Revision** command with the member rule.

To set a member rule, select **Member > Properties > Set Member Rule**.

After you create a rule for a member, you can also view and edit the rule in the **Member Information** view.

---

**NOTE** To create a rule, you require the `ModifyMemberRule` permission.

---

## ***Using a Rule to Update a Member Revision in Another Project Configuration***

By design, applying a rule to update a member revision does not dynamically update the corresponding member revision in another project configuration. For example, if the member revision for `member_1` is updated in `project_A`, the corresponding member revision in a variant of `project_A` with the rule configured to link to `project_A` is not updated with the same member revision.

To update the corresponding member in the variant according to the member rule, you can use the **Update Member Revision** command with the member rule, or your administrator can configure the `ClientLink.js` sample event trigger script that enables dynamic updating of linked member revisions under certain conditions. For more information, contact your administrator.

## Set Member Revision/Rule Dialog Box (GUI)

Tab	Description
<b>Selection</b>	<p>Set a pre-defined or specific revision as the member revision.</p> <p><b>Pre-Defined Revision</b> option is a symbolic revision that represents a location in the development path. To select a pre-defined revision to update the member revision to, click <b>Pre-Defined Revision</b>, then choose a pre-defined revision from the list. Available pre-defined revisions are:</p> <ul style="list-style-type: none"> <li>■ <b>Working</b> updates the member revision to the version of the working file.</li> <li>■ <b>Head</b> updates the member revision to the member's head revision.</li> <li>■ <b>Trunk Tip</b> updates the member revision to the latest revision in the trunk, independent of the default branch settings.</li> <li>■ <b>Member Branch Tip</b> updates the member revision to the latest revision along the member's current branch of development.</li> <li>■ <b>Member Rule</b> updates the member revision to the revision defined as the member rule. The member rule is configured using the <b>Set Member Rule</b> command.</li> </ul> <p>The <b>Specific Revision</b> option allows you to select a revision number, or a revision based on a property (state, label, or timestamp), current project configuration, or external project configuration. To select a specific revision, click <b>Specific Revision</b>, then click one of the following tabs: <b>Revisions</b>, <b>Properties</b>, <b>Project</b>, or <b>Link</b>.</p>
<b>Revisions</b>	Select a specific revision as the member revision.
<b>Properties</b>	<p>Select a revision with a specific state, label, or timestamp as the member revision.</p> <p><b>Last revision with state</b> allows you to make a revision with a specific state the member revision.</p> <p><b>Note:</b> This option displays only if promotion is enabled by your administrator. The options in the list depend on the states configured by your administrator. For more information on the available states, see your administrator.</p> <p><b>Revision with label</b> allows you to make a specific labeled revision the member revision.</p> <p><b>Note:</b> This option displays only if labels exist in the member history.</p> <p><b>Last revision at date</b> allows you to make the most recent revision at the specified timestamp the member revision. Because the revision may be on a different branch closest to the specified timestamp, you can limit the matching of the timestamp to the revision to a <b>Member Branch</b>, <b>Any Branch</b>, or a specific branch in the list.</p>
<b>Project</b>	<p>Select the member revision in another project configuration as the member revision.</p> <p><b>Member revision on Variant</b> allows you to set the member revision to the member revision in a specific variant project.</p> <p><b>Member revision on the Master</b> allows you to set the member revision to the member revision in the master project. This option displays only if you are working in a variant Sandbox or project.</p> <p><b>Member revision on a Project Build</b> allows you to set the member revision to the member revision in a specific build project. You can specify the build project by revision number or label.</p>
<b>Link</b>	<p>Set the member revision to whatever is the member revision for the corresponding member in a specific external project configuration (normal, variant, build). To link the project that the member belongs to (the target project) with the master project, configure the following options:</p> <p>To specify the master project, click <b>Member Revision in a Specified Project</b>. Next, click <b>Select</b> to browse for the specific project and configuration type.</p> <p>Under <b>Member</b>, choose an option to specify the member in the target project:</p> <ul style="list-style-type: none"> <li>■ <b>Find member in specified project</b> searches for the member with the same backing archive in the target project.</li> <li>■ <b>Find member recursively</b> searches recursively throughout the subprojects for the member. There must be exactly one backing archive for each member.</li> <li>■ <b>Specify member name</b> is the name of the member in the target project. This is useful when the target project contains several members that point to the same archive as the current member.</li> </ul>

## Set Member Revision/Rule Options (GUI)

Option	Description
<b>Defer Update Revision</b>	Commits the operation later. This option displays for the <b>Set Member Revision</b> dialog box only. If the change package reviews are mandatory, select this option to create a pending entry for this operation at the time of change package submission. If this option is not enabled, MKS Integrity creates the pending entry at the completion of this procedure.
<b>Close Change Package</b>	Closes the change package when the operation is complete. This option displays for the <b>Set Member Revision</b> dialog box only.
<b>Store Expanded Revision</b>	Sets the rule to the numeric revision number instead of the symbolic revision. For example, if you set the rule according to the <code>ReadyForQA</code> label and select the <b>Store Expanded Revision</b> option, the revision that currently corresponds to the <code>ReadyForQA</code> label is stored and always used when the rule is applied, even if the label later moves to another revision. This option displays for the <b>Set Member Rule</b> dialog box only.
<b>Confirm Rule Clearing</b>	Causes MKS Integrity to confirm that the rule will be cleared. This option displays for the <b>Set Member Rule</b> dialog box only.
<b>Override Rule</b>	Overrides the existing member rule. This option displays for the <b>Set Member Rule</b> dialog box only.

---

# Deleting a Revision

---

**CLI EQUIVALENT** `si deleterevision`

---

If you know you will never use a revision again, you can delete it, provided it is not a pending revision associated with a change package in a state other than `Closed` or `Discarded`, or if change packages are mandatory.

Note the following:

- Only delete a revision when you are certain you will never need it again. Once you delete a revision, it cannot be retrieved. Any historical checkpoints based on a particular revision become invalid if that revision is deleted. A revision cannot be deleted if it is the starting point (root) of a branch. You should never delete the head revision of an archive.
- Any existing locks on revisions are removed when those revisions are deleted.
- If you are using a database type repository, by default MKS Integrity determines if the revision exists in other locations. If the revision is used in other projects, MKS Integrity warns you that deleting the selected revision will break the listed items.

To delete a revision in the GUI, select a revision from a **Member History** view and select **Member > Revision > Delete**.

# Viewing Member Changes By Revision

---

**CLI EQUIVALENT** `si annotate`

---

MKS Integrity provides an annotated revision view for members. Use it when you want to find out which revision introduced a particular change. Rather than searching the content of revisions in the history one revision at a time, you can see the content and information for all of the changes to the member in an annotated list.

For example, Mary needs to know what changes were made to `custom.css`, including by whom and when. She could use the **Member History** view, but that would not show her what lines of code were changed or added. She could also view the differences between each revision one at a time, but that would take too long. The **Annotated Revision** view displays all this information in one window.

In *annotation blocks*, each line of the revision displays with information about the last modification made to each line of the revision's contents. By default, the annotated revision list includes the revision number, author, date, line number, and revision contents.

When using the **Annotated Revision** view:

- you can only view an annotation for one member at a time
- you must close and then reopen the view to see subsequent updates to the member
- only content that was added or changed on a per revision basis displays, but not deleted content

---

**NOTE** The **Annotated Revision** view can only be displayed for members of text format.

---

## Annotated Revision View

In the GUI, select **Member > Views > View Annotated**, or in the Web, select **History > View Annotated Revision**. The **Annotated Revision** view displays.

Revision	Author	Date	Line	Revision Contents
1.2	jrliley	Nov 25, 2009	echo "\nSetting up application build environment"	
1.1	jrliley	Oct 29, 2009	export FINANCIAL_TOOLKIT_ROOT=`pwd`	
1.3	jrliley	Nov 25, 2009	4 echo " NOTE: Type cdaproot to return to cd \$	
1.2	jrliley	Nov 25, 2009	alias cdaproot="cd \$FINANCIAL_TOOLKIT_APP_ROOT"	
1.1	jrliley	Oct 29, 2009	7 cd mytool	
			8 ./setup.ksh	
			9 cd ..	
1.3	jrliley	Nov 25, 2009	10	
			11	
			12 echo "Building java documentation for Source Co	
			13	
			14 cd \$SOURCE_ROOT	
			15	

In the Web interface, the revision information for the specified revision displays in the bottom frame.

---

Select **View** to perform the following tasks:

- **Find** searches for the first instance of a text string in the revision contents column and highlights the text.
- **Find Next** applies the last search to the remaining revision contents column and highlights the next instance the text displays.
- **Find Previous** applies the last search for the text string to the revision contents column in reverse order.
- **Character Encoding** changes the character encoding used to display the revision contents.

**Go to Line** displays a specific line of text. To go to a specific line, enter the number for the line, for example, 33. The line of code displays in the center of the pane if it exists in a scrolling region.

The **Annotated Revision** view display the following information:

Column	Description
<b>Revision</b>	Revision number for each annotated block. If the revision is pending, then pending displays in parentheses.
<b>Author</b>	Author of the revision.
<b>Date</b>	Date each revision was created.
<b>Line</b>	Line number for each line of text in the revision.
<b>Revision Contents</b>	Text contained in each annotation block.
<b>C.P. ID (not displayed by default)</b>	Change package ID for the annotation block.
<b>Label (not displayed by default)</b>	Label for the annotation block.

---

# Working With Pending Revisions

*Pending revisions* are created by MKS Integrity when a change package is submitted. If change package reviews are mandatory, the revision remains pending until the change package is accepted and is committed to the repository. When reviews are not mandatory, you only see pending revisions if a deferred operation that would have created a new revision fails to commit to the repository.

Pending revisions are denoted with *pending* in parenthesis in all views that display revision numbers, for example 1.2 (pending).

In the case of a pending add member operation, the archive name for the pending revision is reserved on the server, thereby preventing other users from specifying the same archive name in any future add member operations.

To review a pending revision created by the action of another user, do one of the following:

- Obtain a working file for the revision by checking out the revision by its number (you cannot obtain a lock on the pending revision).

---

**NOTE** Resyncing a pending revision from your Sandbox replaces the working file (corresponding to the pending revision) with the member revision.

---

- View the working file for the pending change package entry from the **Change Package** view by selecting **Member > Revision > View Contents**.

## **Key Considerations**

- Except where noted below, all commands that involve revisions can be performed on pending revisions.
- Pending revisions:
  - cannot be set as the member revision by a user
  - cannot be the basis for another revision, unless that revision is also a pending revision created by the same user in the same change package
  - can only be locked by their respective creators
  - can only be deleted by discarding the corresponding pending entry from the change package
  - are not recorded in a project checkpoint or Sandbox snapshot, and similarly do not appear in project restore
  - cannot be imported into a project
- When checking out a member that has a pending revision that was created by you, the pending revision is selected by default in the GUI and Web interface if the change package containing the pending operation is open.

# Comparing Revisions

With MKS Integrity you can compare:

- Any two text-based revisions in a history
- Any two symbolic link file revisions in a history
- A text-based revision and its associated working file in a Sandbox (GUI only)
- A symbolic link file revision and its associated working file in a Sandbox (GUI only)
- Any two text files

You can also compare your checked out working file and the locked revision as part of the check in procedure. This enables you to decide whether your changes should be checked in or discarded.

The commands discussed in this section for the GUI use MKS Visual Difference, a tool that compares and merges members and revisions.

MKS Integrity allows you to use a third party differencing tool in the GUI if you do not want to use MKS Visual Difference. You can specify a third party difference tool in your preferences.

---

**NOTE** MKS Visual Difference does not recognize the `cr` line terminator. Consequently the `cr` line terminator may appear as a difference between revisions.

---

## To compare two members in the GUI

- 1 Select two revisions, or a revision and the working file.

Select a working file that has been modified or which has a new revision available.

- 2 Select **Member > Views > View Differences**. Visual Difference launches and displays the two revisions or the working file and revision side-by-side, highlighting the differences between them.

---

**NOTE** A symbolic link is a special type of file that contains a reference that points to another file or directory. When comparing symbolic link file members, the Visual Difference tool does not display. Instead, a dialog box displays the original target file path and the newer target file path.

---

The screenshot shows the MKS Visual Difference tool interface. The title bar reads "MKS Visual Difference - Member AISubsystem.txt". The menu bar includes File, Edit, View, Help. The toolbar contains various icons for file operations. The status bar at the bottom left says "Inserted: 4 | Deleted: 0 | Changed: 0". The main area displays two code panes: "Revision 1.1" on the left and "Revision 1.3" on the right. The code in Revision 1.3 is highlighted in green, indicating inserted lines. The code block is as follows:

```
190     while (count < temp) {  
191         temp--;  
192         Stage = play(new Best(  
193             if (Stage == null) {  
194                 System.out.pri  
195                 UserInput();  
196             }  
197             else {  
198                 System.out.pri  
199                 play(new BEST(  
200                     )  
201             }  
202         System.out.println("Thank you"  
203     })  
204     }  
205     System.out.println("Thank you"  
206 }  
207 }
```

## To compare two revisions in the Web interface

- 1 Select two revisions.
- 2 Select **History > Differences**. Both revisions and a summary of the differences between them are shown in the Differences window.

---

**NOTE** A symbolic link file contains a reference that points to another file or directory. When comparing symbolic link file members, the Visual Differences tool does not display. Instead, a dialog box displays the original target file path and the newer target file path.

---

```

1 echo "\nSetting up application build environment variables..." 1 echo "\nSetting up application build environment variables for FINANCIAL_TO
2                                         _APP..." 2
3 export APPLICATION_ROOT=`pwd` 3 export FINANCIAL_TOOLKIT_ROOT=`pwd`
4 echo "      NOTE: Type cdaproot to return to cd $APPLICATION_ROOT" 4 echo "      NOTE: Type cdaproot to return to cd $FINANCIAL_TOOLKIT_ROOT"
5 alias cdaproot="cd ${APPLICATION_ROOT}" 5 alias cdaproot="cd ${FINANCIAL_TOOLKIT_APP_ROOT}"
6 6
7 cd mytool 7 cd mytool
8 ./setup.ksh 8 ./setup.ksh
9 cd .. 9 cd ..
10
11
12 echo "Building java documentation for Source Code..." 12
13
14 cd $SOURCE_ROOT 14 cd $SOURCE_ROOT
15
16 javadoc -d javadocs -package -classpath "$MYTOOL_CLASSPATH$SEP$FRAMEWORK_CL
17 TH" '$RROOT_DIR/mksnt/find . -name *.java' 16
18
19 //requires error condition 19

```

You can also select from the following rules when viewing the comparison:

- **Ignore blanks** ignores tabs and white space throughout lines in the revisions, but does not ignore the splitting of a word
- **Ignore whitespace** ignores tabs and white space within a line, and ignores the splitting of a word
- **Ignore case** ignores the type case when comparing the revisions
- **Diffs only** displays only the difference blocks in the revisions
- **Wrap** causes the text in the revisions to wrap at a specified character count. Enter the number of characters at which you want the text to wrap within the pane. **Wrap** is enabled by default.

**3** To exit the Differences window, click **OK**.

### To compare two text files in the GUI

- 1 Select **File > Compare Files**. The **Compare Files** dialog box displays.
- 2 Using the **Browse** buttons for each field, select the files you want to compare.

---

**NOTE** The dialog box keeps track of the last 10 files specified in both fields.

---

- 3 Click **OK**. Visual Difference launches and displays the two files side-by-side, highlighting the differences between them, or MKS Integrity informs you there are no differences between the files.

---

# Branching Members

As your projects evolve, you may need to branch members to facilitate development in multiple directions. *Branching* diverges from the mainline within an individual member's history and creates a new revision path. These revisions can then be merged with the member revision on the mainline. *Merging* combines changes into a new revision, usually the head revision.

A branch can start from the mainline or another branch. Revision numbers contain the complete number of the revision that started the branch, for example, 1.2 is branched into 1.2.1.1.

You can branch members while performing a check in operation in the GUI or Web interface by selecting the **Force Creation of New Branch** option.

In addition, members can be branched when:

- you check in a revision using an already existing revision number
- you check in a revision when working in a variant Sandbox

When performing a branch on check in, the member is checked in but is not updated to the member revision. The working revision number increments accordingly, and the original member revision in the project remains unchanged.

# Merging Members

*Branching* diverges from the mainline within an individual member's history and creates a new revision path. These revisions can then be merged with the member revision on the mainline. *Merging* combines changes into a new revision, usually the head revision.

The Merge Branch command merges the entire contents of a branch into a single revision. MKS Integrity keeps track of the changes that have been merged so that later revisions on the branch are not re-merged when the next Merge Branch operation occurs. Any subsequent Merge Branch operations merge only changes since the previous merge operation.

Merging operations are available only in the GUI and CLI.

A merge conflict occurs if the revisions being merged both have changes in a common segment of lines. MKS Integrity provides visual two- or three-way differencing and merging tools. Depending on the operation, merging can launch the MKS Visual Difference tool and the MKS Visual Merge tool to automatically highlight conflicts for resolution. MKS Integrity also supports third-party merging and differencing tools such as Araxis and Beyond Compare.

Even if a no conflict is reported as a result of a merge operation, there can still be inconsistencies in the merge results that can only be detected by someone with an understanding of the context of the change. Therefore, anytime a merge is performed, MKS recommends that you examine and test the merged results before checking them into the repository.

## To merge branched members in the GUI

- 1 Select the revisions you want to merge.

**IMPORTANT** Revisions must be checked in prior to merging. An error occurs if the working file is modified and not checked in.

- 2 Select **Member > Merge > Merge Branch**. The **Merge Branch** dialog box displays.
- 3 Specify the **Target** revision that the branch revision is merged into (typically the member revision on the trunk). You can specify a revision by name from the **Symbolic** list, specify a particular revision from the **Revision** list, or specify a revision by label from the **Label** list.
- 4 Specify the **Branch** you want to merge with the **Target** revision. You specify the branch by specifying a revision on that branch. All of the unmerged changes made on the branch prior to the specified revision are merged with the target revision. You can specify a revision by name from the **Symbolic** list, specify a particular revision from the **Revision** list, or specify a revision by label from the **Label** list.
- 5 To modify the Merge Branch options, click **Options**.
- 6 Under **Change Package**, select a change package to associate with the merged file from the **Change Package** list, or click **Create** to create a new change package for the file.
- 7 To merge the branch into the specified revision, click **OK**. To merge revisions of more than one member, click **OK to All**. MKS Integrity merges all the changes to the working file on the branch (from the point of branching up to the point of check in) into the member revision for the project.

You now have working files in your Sandbox that correspond to the results of the Merge Branch operation; however, they still need to be tested, then checked back into the trunk. Until the changes are checked in, other users cannot see or use them.

---

**TIP** You can also merge branched members in the graphical **Member History** view by dragging the revision at the tip of the branch you want to merge onto the revision that you want to merge into.

---

## Merge Branch Options

Options	Description
<b>Lock Target Revision</b>	Locks the merged revision when the merge is complete using the lock type specified in your locks policy. For information on the locks policy, contact your administrator.
<b>Merge Type</b>	Specify the method you want to use for merging. Select one of the following from the list: <ul style="list-style-type: none"><li>■ <b>Confirm</b> prompts you for the action to be taken.</li><li>■ <b>Cancel</b> cancels the merge operation.</li><li>■ <b>Automatic</b> automatically merges the files.</li><li>■ <b>Manual</b> launches MKS Visual Merge or a third party merge tool, based on your preferences.</li></ul>
<b>On Conflicts</b>	Specify how you want MKS Integrity to handle any conflicts that occur during merging. Select one of the following from the list: <ul style="list-style-type: none"><li>■ <b>Confirm</b> prompts you for the action to be taken.</li><li>■ <b>Cancel</b> cancels the merge operation when a conflict occurs.</li><li>■ <b>Mark For Later Merge</b> exits the merge operation and mark the working file with an unresolved merge icon.</li><li>■ <b>Launch Tool</b> launches the MKS Visual Merge tool.</li><li>■ <b>Highlight Output File</b> completes the merge automatically and highlight the output file where any conflicts occurred.</li><li>■ <b>Error</b> displays an error message prompt.</li></ul>

---

# Merging Modified Working Files

If you modify your working file without having a lock, you can merge your working file changes with the member revision. This allows you to have the most complete working file available.

You can merge your working file automatically when you perform a Check Out or Resync operation on the member by selecting the **Merge Working File If Changed** option.

For example, the member revision of `demoapp.c` is 1.4. Todd has revision 1.3 in his Sandbox and has made changes to the working file. He wants to check out the member revision, but he also wants to merge the changes he made to the working file into the checked out member revision. Todd uses the Automatic Merging on Checkout option.

Depending on your Check Out options or the preferences you have set for the Resync command, MKS Integrity takes one of the following actions when a conflict occurs:

- MKS Integrity asks you to confirm the action to be taken.
- MKS Integrity cancels the operation.
- MKS Integrity exits the merge operation, and marks the working file with an unresolved merge icon.
- MKS Integrity automatically launches the MKS Visual Merge tool.
- MKS Integrity completes the merge automatically and highlights any conflicts in the working file.

When merging on checkout, any changes to the working file are merged and the member is checked out for editing.

When merging on resync, the selected member is updated and merged.

# Resolving Merges

During the process of merging revisions you may decide to suspend the operation and mark the working files for merging at a later time. Working files that are marked for merging appear with an unresolved merge symbol (H) in the delta column in the **Sandbox** view.

If you have begun the merging process but decided to use the **Mark for Later Merge** option to postpone the completion of the merge process, when you are ready to merge the selected revisions you can use the **Resolve Merge** command to do so.

---

**NOTE** You cannot suspend a merge operation and mark the working files for merging at a later time when using a third party merge tool.

---

## To resolve a merge in the GUI

- 1 Select a member with an unresolved merge symbol (H).
- 2 Select **Member > Merge > Resolve Merge**. The **Merge** dialog box displays.
- 3 Select one of the following options for how you want to complete the merge, then click **OK**:
  - **Automatically** completes the merge process without launching the MKS Visual Merge tool.  
If MKS Integrity encounters a conflict, depending on your preferences the operation may be canceled, the revisions may be marked for merging later, the output file may be highlighted, or the MKS Visual Merge tool may be launched.
  - **Manually** allows you to complete the merge operation through MKS Visual Merge or a third party merge tool, based on your preferences. The merge tool appears in a new window displaying the revisions you want to merge. Perform merging and editing as required, then save the merge results and close the tool.

In your **Sandbox** view, a delta displays indicating that the merge is complete.

# Working With MKS Visual Difference

The MKS Visual Difference tool is a graphical application that allows you to compare revisions. It offers two-way differencing of revisions where differences are highlighted for you. MKS Visual Difference operates in two modes, difference mode and merge mode. You must be in merge mode to merge revisions in MKS Visual Difference.

**IMPORTANT** Members must be checked out before you can merge them.

**TIP** You can configure the character encoding used to display the revision contents in the tool.

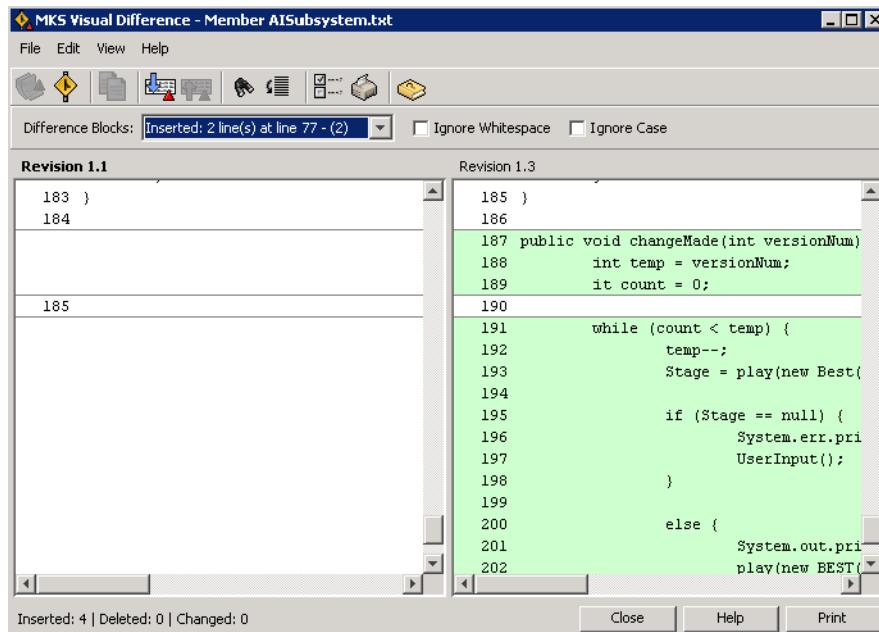
**CAUTION** Changing character encoding during the merge operation is not supported. A confirmation message notifies you that the current merge result will be lost if you continue with the character encoding change.

## Merging Two Revisions

You can use MKS Visual Difference to merge differences between two revisions selected from a member history or a revision and a working file. The GUI and CLI allow you to merge differences between two revisions.

### To merge two revisions in the GUI

- 1 Compare two revisions whose differences you want to merge.
- 2 In Visual Difference, select **File > Merge**. The **Reassign Merge Roles** dialog box displays.
- 3 Select from the list the revisions you want as the **Merge From** revision and the **Merge To** revision.
- 4 Click **OK** to continue to Merge mode. MKS Visual Difference switches to the merge mode split layout.



- 
- 5** Complete the required changes by doing one of the following:
    - merging blocks (see “Merging Blocks” on page 386)
    - editing merge results (see “Editing Merge Results” on page 386)
  - 6** To save the **Merge Result**, select **File > Save As** (or **File > Save** if you have already specified a file name for the merge result). The **Save merge result** dialog box displays.
  - 7** Type the file name you want to save your merge result as. By default, MKS Integrity selects the file name corresponding to the member name.
  - 8** Click **Save** to save the **Merge Result**.

## Merging Blocks

Difference blocks are highlighted within each revision and may be insertions, deletions, or changes. You can select the blocks you want to merge by this method.

The merging by blocks functions are accessible from the **Edit** menu and the shortcut menu. The following outlines the operations you can use:

- **Merge Block** replaces a block in the **Merge Result**. The block you select to merge is inserted, replacing the corresponding block in the **Merge Result**.
- **Merge Block Above** inserts the selected block above the adjacent block in the merge result file.
- **Merge Block Below** inserts the selected block below the adjacent block in the merge result file.

You can also perform these operations in the following ways:

- Double-click a block in the **Merge From** or **Merge To** panes to replace the adjacent block in the **Merge Result**.
- Drag blocks from the **Merge From** and **Merge To** panes to the **Merge Result** pane.

You must save the merge result file to complete the merging operation. For information on saving merge results, see “Saving Merge Results” on page 387.

## Editing Merge Results

Once you are working in merge mode in MKS Visual Difference, you can directly edit the merge result if necessary.

In MKS Visual Difference you can cut, copy, and paste text in the merge result. MKS Visual Difference highlights the edited text and displays the edit icon (edit icon) next to the line number. You can access editing functions in the following ways:

- From the **Edit** menu, select **Cut**, **Copy**, **Paste**, or **Select All**.
- From the toolbar, click the **Cut**, **Copy**, or **Paste** buttons.
- From the shortcut menu, select **Cut**, **Copy**, **Paste**, or **Select All**.
- Use the following shortcut keys:
  - **Cut** = **CTRL+X**
  - **Copy** = **CTRL+C**
  - **Paste** = **CTRL+V**
  - **Select All** = **CTRL+A**

---

You can also add and delete text by selecting a line within the merge result and type or delete as required.

## Reverting Merge Results

If you have made changes to the merge result file that you want to discard, you can revert the merge result file to its last saved state or revert a particular block.

Operation	Procedure
To revert merge results in the GUI	Select <b>Edit &gt; Revert Merge Result File</b> .
To revert merge results by block in the GUI	Select the block you want to revert in the merge result file and select <b>Edit &gt; Revert Block</b> .

## Saving Merge Results

Once you have edited or merged blocks into the merge result file, you must save the file. You can save the file as the member and check it in to continue development or you can choose a file name of your choice. If you do not save the merge result file as the member, MKS Integrity does not recognize any changes to the member.

To save merge results in the GUI, select **File > Save As**, type the file name in the **Save merge result** dialog box, and click **Save** to save the merge result file.

By default, if the working file is involved in the comparison, MKS Integrity selects the file name corresponding to the member name.

---

**NOTE** If you save the merge result to your working file, MKS Visual Difference automatically recalculates the differences when it is saved.

---

Once you have selected a file name and saved the merge result file, as you continue to make changes you can select **File > Save** to save it.

# Working With MKS Visual Merge

The MKS Visual Merge tool is a graphical application that allows you to compare, edit, and merge revisions. It offers three-way differencing of revisions where conflicts are highlighted for you.

**TIP** You can configure the character encoding used to display the revision contents in the tool.

**CAUTION** Changing character encoding during the merge operation is not supported. A confirmation message notifies you that the current merge result will be lost if you continue with the character encoding change.

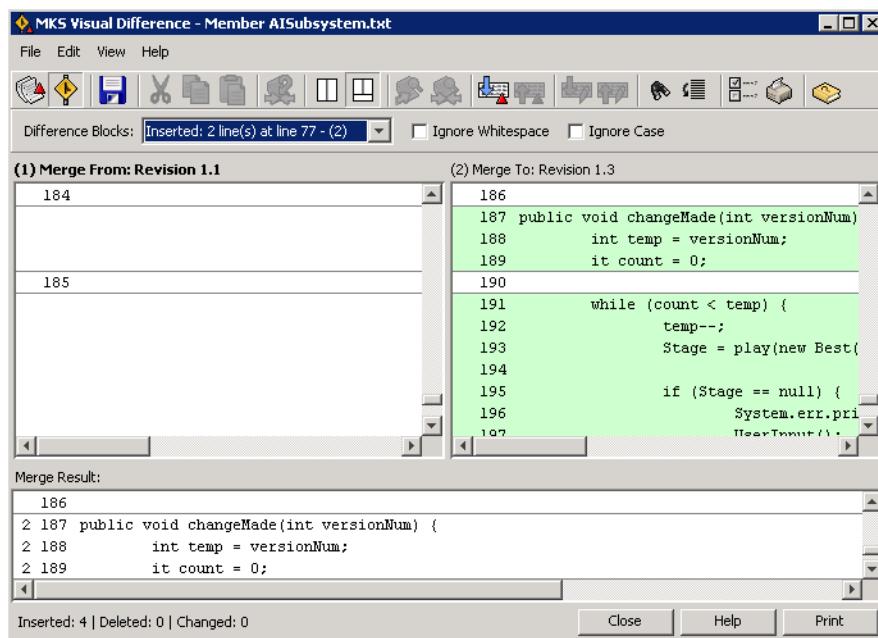
## Viewing Conflicts

Conflicts, if they exist, are highlighted and appear with a red flag next to the line number in the view panes. Conflicts are also listed in the **Difference Blocks** list. You can decide which of the conflicting blocks you want to incorporate into the merge result, and use the merging and editing utilities to revise the merge result.

**NOTE** To minimize conflicts, only the \$Log\$ keyword is expanded in the MKS Visual Merge tool, all other keywords unexpanded. The \$Log\$ keyword cannot be unexpanded because it appends information rather than replaces it. The \$Log\$ keyword appears as a difference, but not a conflict.

You can use the **Next Conflict Block** (➡) and **Previous Conflict Block** (⬅) buttons to navigate through the revisions and view all the conflicts.

**NOTE** MKS Visual Merge does not recognize the cr line terminator. Consequently the cr line terminator may appear as a conflict or difference between revisions.



---

## Merging Nonconflicting Blocks

Merging nonconflicting blocks allows you to merge without including any areas of conflict. MKS Visual Merge incorporates all difference blocks (except conflict blocks) from the **Merge From** revision into the merge result file.

To merge all nonconflicting blocks in MKS Visual Merge, select **Edit > Merge All Nonconflicting Blocks**. All nonconflicting difference blocks are applied to the merge result file.

Save the merge result file to retain the changes as described in “Saving Merge Results” on page 390.

## Merging Individual Blocks

If you do not want to merge entire revisions you can merge individual difference blocks or lines. Difference blocks are highlighted within each revision and may be insertions, deletions, changes, or conflicts.

The merging blocks functions are accessible from the **Edit** menu and the shortcut menu. The following outlines the commands you can use:

- **Merge Block** inserts the selected block or line into the merge result file in the same location as it displays in the revision it originated from.
- **Merge Block Above** inserts the selected block or line above the adjacent block in the merge result file.
- **Merge Block Below** inserts the selected block or line below the adjacent block in the merge result file.

In addition to these commands, in MKS Visual Merge you can simply select a block and double-click it or drag it into the merge result file.

You must save the merge result file to complete the merging operation. For information on saving merge results, see “Saving Merge Results” on page 390.

## Editing Merge Results

While you are working in MKS Visual Merge, you can directly edit the merge result if necessary. This is typically required for resolving merge conflicts, but can be used in other situations as well.

In MKS Visual Merge you can cut, copy, and paste text in the merge result. MKS Visual Merge highlights the edited text and displays the edit icon (edit icon) next to the line number. You can access editing functions in the following ways:

- From the **Edit** menu, select **Cut**, **Copy**, **Paste**, or **Select All**.
- From the toolbar, click the **Cut**, **Copy**, or **Paste** buttons.
- From the shortcut menu, select **Cut**, **Copy**, **Paste**, or **Select All**.
- Use the following shortcut keys:
  - **Cut** = **CTRL+X**
  - **Copy** = **CTRL+C**
  - **Paste** = **CTRL+V**
  - **Select All** = **CTRL+A**

---

You can also add and delete text by selecting a line within the merge result and type or delete as required.

## Reverting Merge Results

If you have made changes to the merge result file that you want to discard, you can revert the merge result file to its original state or revert a particular block or line.

To revert merge results to the original state in MKS Visual Merge, select **Edit > Revert Merge Result File**. The merge result file reverts to its last saved state.

To revert merge results by block in MKS Visual Merge, select the block or line you want to revert in the merge result file, and select **Edit > Revert Block**. The block or line is removed from the merge result file.

## Saving Merge Results

Once you have edited or merged blocks into the merge result file, you must save the file. You can save the file as the member and check it in to continue development, or you can choose a file name of your choice. If you do not save the merge result file as the member, MKS Integrity does not recognize any changes to the member.

To save merge results in MKS Visual Merge, select **File > Save As**, type the file name in the **Save merge result** dialog box, and click **Save**.

---

**NOTE** If you save the merge result to your working file, MKS Visual Merge automatically recalculates the differences when it is saved.

---

Once you have selected a file name and saved the merge result file, as you continue to make changes you can select **File > Save** to save it.

## Suspending Merges

Once you have initiated a merge, you can suspend it and then resolve it at a later time. Suspending merges allows you to mark the working file to indicate that merging is required. This provides time you may need to investigate conflicts or difference blocks before finishing the merge.

You can only suspend a merge operation if there are changes since the last time you saved the merge result.

To suspend a merge, in MKS Visual Merge, click the **Close** button on the status bar, then click **Yes** to mark the working file for merging at a later time. In your **Sandbox** view, the member is marked with an unresolved merge symbol ()<sup>1</sup>, and in the member information pane MKS Integrity indicates that resolution is required.

---

# Deferring Member Operations

For certain operations in MKS Integrity, you can defer the completion of the command until a later time. The defer option allows you to see the effect of the operation in your Sandbox without affecting the project. The deferral of operations is provided as a selectable option in two ways: through the GUI and in the command line interface as the `--defer` option.

For example, Chad is assigned to change the current savings calculator menu image to match the other menu images. To do this, he needs to drop the current image and add a new one. Chad uses a deferred add and drop because he does not want to commit his changes until he finishes unit testing his modifications.

You can apply the defer option to the following operations:

- Adding members to a project
- Adding members from archive to a project
- Moving members
- Dropping members from a project
- Checking in a member
- Renaming a member
- Importing members
- Viewing an annotated revision

---

**TIP** You can apply the **Deferred Items** filter to display any members that are associated with deferred operations.

---

A *deferred member* is a member that is associated with any deferred operation (add, drop, checkin, rename, move, import, add from archive, update revision). A deferred member displays in the Sandbox, but the deferred operation is not shown in the project until the deferred operation is submitted.

Deferred operations are visible only from the client-side Sandbox and are seen in the project only when they are submitted. It is important to note that deferred operations can be added to a change package and then submitted as a group of changes, but the deferred operations do not appear on the MKS Integrity Server until they are submitted to the project.

With the exception of deferred rename, move, and checkin operations, MKS Integrity supports only single deferred operations on a member. If you try to perform multiple deferred operations on a single member, MKS Integrity reports an error.

---

**NOTE** Once you have specified a deferred operation on a member, MKS Integrity does not allow any further operations that would cause that member to be modified.

---

You can simultaneously defer rename, move and/or checkin operations on a single member. When these deferred operations are submitted, MKS Integrity first performs the checkin operation followed by the rename operation, then the move operation. The sequence for performing these operations is set by default and is not configurable.

If change package reviews are mandatory, ensure that the deferred option is enabled in command dialog boxes, or MKS Integrity creates pending entries (and if necessary pending revisions) at the time the command operation is completed rather than when the associated change package is submitted.

---

# Generating Member or Project Reports

---

## CLI EQUIVALENT `si report`

---

If you have workflows and documents enabled on your system and your administrator has set up MKS Integrity items to collect MKS Integrity configuration management project information, you can use the expanded reporting capabilities provided by MKS Integrity.

If you do not have this functionality enabled, you can use the `si report` command from the CLI with an alternative reporting tool to report on configuration management project information.

MKS Integrity Reporter analyzes configuration management projects, their members, or individual archives, and allows you to generate a variety of reports and graphs based on its findings. For example, you could create a report to show the changes that have been made to a source file, and which team members made them.

Reporter calculates a summary of the changes to a project or archive, then displays or prints the summary as text or, for some reports, optionally as a graph. If you display a report as a graph, you can define how you want it to look to suit your preferences or for quick, project-specific identification. You can save Reporter's data files (containing the results of its project or archive analysis) as text files that most database applications can read.

MKS Integrity Reporter outputs the report as CSV formatted files containing report data.

Reporter generates the following summaries:

- changes introduced by individual authors
- changes between the member revision and a revision with a particular label
- changes between the member revision and any other revision
- a list of locked members and the names of users who locked them
- a list of revisions with a particular label or state
- project member history (including revision descriptions) by file

For more information on using the `si report` command, see the *MKS Integrity 2009 CLI Reference Guide for Configuration Management*.

# Promoting and Demoting Members

When MKS Integrity is used on its own, your administrator can define states that members move through as part of the development process.

A state is free-form text used to classify the condition of a revision in a member history. For example, a document could initially have a state of "Draft". As work progresses it might be changed to "Review" and eventually "Complete". If no state is assigned to a revision, a default value of "Exp" (for Experimental) is used.

To control member workflow more effectively, MKS recommends you use software configuration management with process and workflow management.

## Promoting Members

---

### CLI EQUIVALENT `si promote`

---

Promotion is the process of managing data as it moves through a structured development cycle. Each phase is represented by states that are defined by the administrator. Project members can also be demoted to a predefined lower state.

For example, Patrick the project manager is notified that the ABC Financial Toolkit has reached code freeze. He promotes the project and members to `Testing` and notifies Quality Assurance that they can begin final testing of the toolkit.

---

**NOTE** Promoting members is for historical purposes only. To control member workflow more effectively, MKS recommends you use MKS Integrity.

---

At particular milestones, project members are ready to move to the next stage of their development cycle (for example, from `Development` to `Test`). If no state system is defined, a default value of `Exp` (Experimental) is assigned to all revisions.

Interface	Procedure
GUI	Select <b>Member &gt; Properties &gt; Promote</b> . In the <b>Promote Member</b> dialog box, select a new state from the <b>Promote to State</b> list.
Web	Select <b>Member &gt; Promote</b> . In the <b>Promote Member</b> dialog box, select a new state from the <b>Promote to State</b> list.

## Demoting Members

---

### CLI EQUIVALENT `si demote`

---

---

Project members can also be demoted to a predefined lower state of development.

---

**NOTE** Demoting members is for historical purposes only. To control member workflow more effectively, MKS recommends using MKS Integrity.

---

Interface	Procedure
GUI	Select <b>Member &gt; Properties &gt; Demote</b> . In the <b>Demote Member</b> dialog box, select a new state from the <b>Demote to State</b> list.
Web	Select <b>Member &gt; Demote</b> . In the <b>Demote Member</b> dialog box, select a new state from the <b>Demote to State</b> list.

---

## PART 12

# Grouping Units of Work in Change Packages

---

# Change Package Overview

A *change package* is a group of changes made by a single user that can be considered a logical unit of work. Only the creator of a change package can add entries to that change package. Change package entries are added when you specify a change package while performing member operations.

A change package acts as a log of both the changes to members and subprojects that have already been committed to the repository (server), and the changes that are only visible to the user on the desktop and not committed to the repository (deferred). When change package reviews are mandatory, a change package acts as a control placed on changes to the repository by making them pending before they are committed.

A change package is open until you close it, which signifies that work on the change is completed. When reviews are mandatory, a change package has additional states before it is closed.

The following rules apply when using change packages:

- Each change package has a unique change package ID (CP ID). The CP ID is a colon separated identifier of the form:

```
<container ID>:<relative change package number>
```

---

**NOTE** If the MKS Integrity integration is enabled, the item ID is used as the container ID.

---

- Only the creator of a change package or a change package administrator can close a change package.
- Once a change package is closed, it can only be re-opened by a change package administrator. If a change package has been propagated through Apply CP or Resync CP, it cannot be reopened, even by a change package administrator.

You can expand the capabilities of change packages by associating them with items to take advantage of MKS Integrity's workflow and document management.

## **Integrating With Workflows and Documents**

As part of integrating with workflows and documents, you can associate change packages with MKS Integrity items. Items provide more detailed information on the action that is required and control the workflow of development and testing, known as the software development cycle.

MKS Integrity items track changes in the software development cycle. For example, your administrator can configure MKS Integrity in a way that a *problem* item may be associated with a *solution* item for easy tracking and monitoring of both items. Each item has an audit trail, which may be used to evaluate internal processes for the effectiveness of the problem resolution process. In effect, items track all aspects of any engineering project.

Integrating with workflows and documents allows you to specify groups of developers who are affected by an item, who can then be notified using e-mail notification rules.

Administrators define what item types can use change packages and configure software configuration management to integrate with process and workflow management.

Change packages can be grouped together using an item to create a larger logical grouping of changes. Each change package in an item can be created by a different user for team development of an item.

Note the following rules that apply when using items and change packages:

- 
- Multiple change packages can be created for a single item. If the resolution of an item requires more than one set of changes, a new change package can be created for each new set of changes. This also allows multiple users to work on the same item.
  - If an item type does not allow open change packages, you cannot create and associate new change packages with that item. Check with your administrator to find out which item types allow open change packages.
  - Only item states specified by your administrator allow open change packages. When attempting to advance to a state that does not allow open change packages, MKS Integrity instructs you to first close the item's change package.
  - Typically, an item cannot advance to the final state in an MKS Integrity workflow until all change packages are closed. See your administrator for more information.
  - When workflows are enabled, the change package ID is a colon-separated identifier of the form:  
`<item number>:<relative change package number>`

### **Why Use Change Packages?**

Change packages provide the following advantages:

- Change packages allow related changes to be grouped as a logical unit.
- Change packages allow work in progress to be submitted to the repository (server) all at once (using submit change package), which prevents users from partially submitting related work in progress.
- Change packages provide a way to apply related changes to a project or Sandbox in one operation.
- Using change packages, users are able to resync the changes required to address a specific item without resyncing the entire project.
- Groups of changes can be reviewed as a unit. If reviews are mandatory, changes are reviewed before they are committed to the server repository.

### **Using Change Packages to Control Development**

The following is an example of a way to use change packages to control development. You can combine this example with the review process to control what changes become a permanent part of the repository.

- 1 If you have workflows and documents enabled, submit an item that needs to be addressed.
- 2 Create a change package for the group of tasks you need to perform to address an item. When creating the change package associate it with the item you created.
- 3 MKS Integrity assigns the change package an ID and leaves the change package in an open state. You can see the change package listed in the **Change Packages** view.
- 4 As part of your development process, identify the members that are affected by the item. Specify the change package when performing operations on the affected members. The operations are listed in the **Change Package** view.
- 5 When all of the development to address the item is completed, submit the change package. Any locked members are converted to deferred checkin operations, then checked in. All of the deferred operations are submitted.
- 6 Close the change package when the work to address the item is completed. The change package is moved to the **Closed** state, and the change package disappears from the **Change Packages** view.

- 
- 7** Advance the item through the workflow. At this point, the verification phase begins. If it is determined that more work needs to be performed to address the item, the user moves the item to an earlier state in the workflow, then you (or another developer) create an additional change package. The process is repeated until the item is sufficiently addressed.

---

# My Change Packages View: Managing Change Packages

To display the **My Change Packages View**, select **Change Package > View My Change Packages**.

MKS Integrity provides a central location to manage change packages you have created that contain entries that have not been committed to the server repository. From one location, you can view a change package and its entries, view any items associated with a change package, submit any deferred operations for that change package, close a change package, edit a change package, or create a new change package.

If reviews are mandatory, submitted change packages continue to appear in the **Change Packages** view until they are successfully committed to the repository.

The **My Change Packages** view and **Filtered Change Packages** view display information in the same way. The **My Change Packages** view only displays your open change packages. To display information for other change packages in the GUI, select **Change Package > Find** and specify filter criteria to display the **Filtered Change Packages** view.

By default, the **My Change Packages** view displays the following information:

Column	Description
<b>ID</b>	Change package ID.
<b>Item</b>	Item ID if MKS Integrity is enabled.
<b>Creator</b>	User name of the person who created the change package.
<b>State</b>	Current state of the change package
<b>Summary</b>	Summary statement for the change package.

# Creating a Change Package

---

**CLI EQUIVALENT** si createcp

---

You can create a change package in MKS Integrity in one of the following ways:

- Select **Change Package > Create**.
- When performing a member operation, in the **Change Package** portion of the dialog box that displays, click **Create**.

If workflows are enabled, when creating a change package you can choose an MKS Integrity item to associate with the change package.

Your ability to create a change package associated with an item is based on the change package creation policy for the item type.

## **Working With Change Package Entries**

Once a change package is created, you can add entries to that change package by selecting it in the **Change Package** portion of the dialog box or wizard when performing member and subproject operations. To add change package entries to a change package, the MKS Integrity Client must be connected to the server the change package resides on.

When performing an operation that can add entries to a change package, keep the following in mind:

- A change package can only be specified if your administrator has enabled the use of change packages.
- You must specify a change package if your administrator has made change packages mandatory.
- If change packages are not mandatory, and if no change package is applicable, you can select **<none>** in the **Change Package** field.
- If change packages are mandatory, depending on change package configuration set by your administrator, you can bypass the change package requirement by selecting **bypass** in the change package field.

---

**TIP** You can also add an entry to a change package by dragging the member from a **Sandbox** view to a **Change Package** view (or change package in the **Change Packages** view). The **Check Out** dialog box displays with the change package selected.

---

After the operation is completed, the entry is added to the change package. Depending on the operation, the change package's ID displays in one of the member's CPID (change package ID) columns in the **Project** or **Sandbox** view. Not all CPID columns appear by default.

If the change package contains deferred operations or lock entries, a  icon displays beside the change package ID in the **Working CPID** column of the **Sandbox** view.

## **Change Package Entry Types**

The following table displays change package entry types and the operation that creates them.

Change Package Entry Types	Operation
Add	Add Member
AddFromArchive	Add Member From Archive

---

Change Package Entry Types	Operation
AddSubproject	Add Subproject
AddSharedSubproject	Add Shared Subproject
ConfigureSubproject	Configure Subproject
CreateSubproject	Create Subproject
Drop	Drop Member
DropSubproject	Drop Subproject
Import	Import Member
Lock	Lock Member, Check Out <b>Note:</b> This entry is used for exclusive locks.
NonExclusiveLock	Lock Member, Check Out
MoveMember	Move Member
MoveSubproject	Move Subproject
Rename	Rename Member
Update	Check In (with updating member revision)
UpdateRevision	Update Member Revision
UpdateArchive	Check In (without updating member revision)

# Discarding ChangePackage Entries

You can remove entries from a change package by discarding them. To discard entries, the change package must be in an Open or Rejected state.

You can only discard entries from a change package in a Closed state if all of the following conditions are met:

- You are the change package creator or administrator.
- Reviews are not mandatory for the change package. If change packages are mandatory, committed entries cannot be discarded (contact your administrator for more information).
- The change package has not been propagated by another change package

Interface	Procedure
GUI	From the <b>Entries</b> panel on the <b>Change Package</b> view, select the change package entries you want to discard. Select <b>Change Package &gt; Entry &gt; Discard</b> .
Web	From a <b>Project</b> or <b>Member History</b> view, click the change package ID. From the <b>Change Package</b> view, click the <b>Entries</b> tab. On the <b>Entries</b> panel, select the entries you want to discard and select <b>Actions &gt; Discard Change Package Entry</b> .

When a change package entry is discarded, the following happens if applicable:

- Deferred operations corresponding to deferred entries are reverted.
- Locks on members corresponding to lock entries are released.
- Pending operations corresponding to pending entries are reverted, and discarded entries are created (to preserve the review history).
- Pending revisions corresponding to pending entries are deleted.
- Archives created for pending members associated with entries in the change package are deleted from the server (pre-existing archives that were shared to create a pending member are not deleted).
- Any nested operations in a pending subproject are discarded at the same time as the subproject.

---

# Viewing ChangePackage Entry Member Differences

From the **Change Package** view, you can view the member differences of any change package entry.

Interface	Procedure
GUI	From the <b>Entries</b> panel on the <b>Change Package</b> view, select the change package entry you want to difference. Select <b>Member &gt; Views &gt; View Difference</b> .
Web	From a <b>Project</b> view, select a member that has a change package associated with it. Under the <b>C.P.ID</b> column, click the number link for the change package. From the <b>Change Package</b> view, select the change package entry you want to difference and select <b>Actions &gt; View Member Differences</b> .

For updates, MKS Integrity compares the most recent revision checked into the change package with its immediately preceding revision. For example, if you want to view the differences for the member `readme.txt` at revision `1.3`, MKS Integrity compares revisions `1.3` and `1.2`, displaying them in the Differences window.

For lock entries in the GUI, MKS Integrity compares the working file with the locked revision. You cannot difference a lock entry that does not have an associated Sandbox.

The Differences window displays the two revisions side-by-side, highlighting the differences between them, or MKS Integrity informs you that there are no differences.

---

# Finding Change Packages

MKS Integrity provides you with a way to search through all of the change packages created by anyone, including closed change packages.

To find change packages in the GUI, select **Change Package > Find**.

To find change packages in the Web, create a query.

In the GUI, you can use the the following search methods:

- **By Filter**

To combine filters, select the desired filters, then select **Logical AND** or **Logical OR** to specify their relationship.

To invert a filter, click the filter a second time and the **!** symbol displays. For example, you can search for change packages that are not associated with a specified project.

- **By ID**

Specify a container ID, for example `12`, finds all changes packages with that container ID. If the workflows are enabled, the container ID is the same as the item ID.

Specify the full change package ID, for example `12:1`, finds the single change package.

You can search by as many IDs as you want.

- **By Query**

If workflows are enabled, you search for change packages by query. This enables you to find change packages based on complex criteria.

---

**NOTE** If you use a query that contains a symbolic date (for example, `today`), your query results may be different than if you ran the same query in MKS Integrity. This is because the MKS Integrity workflow and document management functionality uses the client's time zone while the MKS Integrity configuration management functionality uses the server's time zone.

---

## Find Change Packages Options

Options	Description
<b>State</b>	Finds change packages if they are currently in the specified state.
<b>Type</b>	Finds change packages that are of the specified type.
<b>Created By</b>	Finds change packages that were created by a specified user.
<b>Creation Date</b>	Finds change packages that were created on a specified date. Default date is <a href="#">today</a> . Click <b>Change</b> to specify another date using the <b>Creation Date Filter</b> .
<b>Closed Date</b>	Finds change packages that were closed on a specified date. Default date is <a href="#">today</a> . Click <b>Change</b> to specify another date using the <b>Creation Date Filter</b> .
<b>Member Name</b>	Finds change packages that contain a specified member name.
<b>Member Type</b>	Finds change packages that contain entries for member operations, subproject operations, or both.
<b>Project</b>	Finds change packages whose members belong to a specified project.
<b>Variant</b>	Finds change packages whose members belong to a specified project variant.
<b>Summary</b>	Finds change packages that contain specified text in the change package summary.
<b>Description</b>	Finds change packages that contain specified text in the change package description.
<b>Entry Type Modifier</b>	Finds change packages based on their category. Categories are <a href="#">Committed</a> and <a href="#">Pending</a> .
<b>Entry Type</b>	Finds change packages that contain a specified entry type. <b>Entry Type</b> filter only finds lock entries that were made from a <b>Project</b> view.
<b>Staging System</b>	Finds change packages that are being deployed in a specified staging system. For more information, see the <a href="#">MKS Deploy 2009 Administration Guide</a> .
<b>Stage</b>	Finds change packages that are in a specified stage of the staging system. For more information, see the <a href="#">MKS Deploy 2009 Administration Guide</a> .
<b>Deploy Target</b>	Finds change packages that have a deploy target. For more information, see the <a href="#">MKS Deploy 2009 Administration Guide</a> .
<b>Deploy Request State</b>	Finds change packages that have a deploy request in a specified state. For more information, see the <a href="#">MKS Deploy 2009 Administration Guide</a> .
<b>Deploy Request ID</b>	Finds change packages that have a deploy request with a specified ID. For more information, see the <a href="#">MKS Deploy 2009 Administration Guide</a> .
<b>Pending Review By</b>	Finds change packages submitted for review but not yet accepted or rejected by a specified reviewer.
<b>Accepted By</b>	Finds change packages accepted by a specified user.
<b>Rejected By</b>	Finds change packages rejected by a specified user.
<b>Associated with Item</b>	Finds change packages that are associated with an MKS Integrity item.

# Viewing Change Package Details and Entries

**CLI EQUIVALENT** `si viewcp, si viewcps`

Once you add entries to a change package, you can use MKS Integrity to view the change package information for the member or subproject.

Operation	Procedure
To view information for a change package in the GUI	Select a change package in the <b>Change Packages</b> view, and then select <b>Change Package &gt; View Change Package Details</b>
To view information for a change package associated with a member operation in the GUI	Select the member in a <b>Project</b> or <b>Sandbox</b> view, and then select <b>Member &gt; Change Package</b>
To view information for a change package associated with an annotated member revision in the GUI	From the <b>Annotated Revision</b> view, select the annotation block corresponding to the change package you want to view, and then select <b>Change Package &gt; View Change Package Details</b>
To view information for a change package associated with a subproject operation in the GUI	Select the subproject in a <b>Project</b> or <b>Sandbox</b> view, and then select <b>Project &gt; Subproject Change Package</b>
To view information for a change package in the Web interface	Select a member or revision in the <b>Project</b> or <b>Member History</b> view, and then select <b>Change Package &gt; View Change Package Details</b> <b>Tip:</b> You can also click the change package ID in the <b>Project</b> or <b>Member History</b> view to view the change package.

When viewing information for change packages associated with a member operation, you can select one of the following:

- **View Working** displays the change package associated with a deferred or lock operation performed by the current user from the current Sandbox (available only through a Sandbox context).
- **View Member** displays the change package associated with the operation that set the member revision.
- **View Creation** displays the change package that created the revision that is currently the member revision. This revision may be different from the member change package ID if you used an import, add member from archive, or set member revision operation.
- **View Locker** displays the change package associated with the non-exclusive lock on the member revision. If you use a non-exclusive locking policy, there can be multiple locks on the member revision. If there are multiple locks, the current user's lock takes priority. If the current user does not have a lock associated with a change package, the change package associated with an exclusive lock displays. If there is no exclusive lock associated with a change package, the first non-exclusive lock on the member revision displays.
- **View Pending** displays the change package associated with a pending operation.

When viewing information for change packages associated with a subproject operation, you can select one of the following:

- 
- **View Member** displays the change package associated with the current subproject configuration.
  - **View Pending** displays the change package associated with a pending subproject operation.

## Change Package View

By default, the **Change Package** view displays the following information:

Panel	Description
<b>Attributes</b>	<p><b>Attributes</b> panel displays the change package ID, server the change package was made on, change package type, summary, current change package state, person who created the change package, date the change package was created, and description (if one was provided).</p> <p>If applicable, the <b>Attributes</b> panel also displays the closed date and resolution information.</p> <p>If the change package is a staging and deploy change package, the <b>Attributes</b> panel also displays the staging system that the change package is being deployed in, the stage of the change package in the staging system, the deploy target for the change package, the state of the deploy request for the change package, and the ID of the deploy request for the change package. For more information, see the <i>MKS Deploy 2009 Administration Guide</i>.</p>
<b>Entries</b>	<p><b>Type</b> is the entry type of the change package entry.</p> <p>In the Web interface, clicking an entry type displays revision differences associated with the entry in MKS Visual Difference.</p> <p><b>Member</b> displays the name of the member or subproject affected by the operation.</p> <p>When it is a <b>Rename</b> entry type, the member name displays with the form: newname (oldname). When it is a <b>MoveMember</b> or <b>MoveSubproject</b> entry type, the member name displays in the form: newlocation (oldlocation).</p> <p><b>Revision</b> displays the number of the revision in the change package entry.</p> <p><b>Sandbox</b> displays the name of the Sandbox where the deferred operation or checkout took place.</p> <p><b>Project</b> displays the name and path of the project where the operation was performed. If the operation occurred in a shared subproject, the project where the subproject is shared from is displayed. If the operation involved two different projects (for example, moving a member from one project to another), the source project is displayed for deferred operations, and the target project is displayed for pending or committed operations.</p> <p><b>Variant</b> displays the name of the variant development path (if a variant was used) the change package entry occurred on.</p> <p><b>Date Changed</b> displays the date the entry was made.</p> <p><b>Server</b> displays the host name of the server the entry was made on.</p> <p>In the GUI, additional columns are available in the client preferences.</p>

Panel	Description
<b>Review Log</b>	<p>If reviews are mandatory, the <b>Review Log</b> panel displays review information. Each change package review log contains a record for each reviewer of the change package. Each review record contains the following information:</p>
	<p>Possible review states are:</p> <ul style="list-style-type: none"> <li>■ <b>Review Pending:</b> the change package is still in a state of <b>Submitted</b> and there are outstanding votes to be cast by reviewers.</li> <li>■ <b>Accepted:</b> the change package is in a state of <b>Accepted</b>, and all of the individual reviewers and a user from each reviewer group have accepted the change package.</li> <li>■ <b>Rejected:</b> the change package is in a state of <b>Rejected</b>, and at least one reviewer has rejected the change package.</li> </ul>
	<p>Possible reviewer types are:</p> <ul style="list-style-type: none"> <li>■ <b>User Reviewer</b> is a user voting in an individual capacity.</li> <li>■ <b>Group Reviewer</b> is a user voting in a group capacity on behalf of that group.</li> <li>■ <b>Super Reviewer</b> is a user casting an overriding vote in a super reviewer capacity, and is not required to be a user or in a group on the reviewer list.</li> </ul>
	<p>Possible vote values are:</p> <ul style="list-style-type: none"> <li>■ <b>Pending</b> signifies that the reviewer or group reviewer has not yet cast a vote.</li> <li>■ <b>Accepted</b> signifies that the user has cast an accept vote for the change package.</li> <li>■ <b>Rejected</b> signifies that the user has cast a reject vote for the change package.</li> </ul>
	<p>Displays information that reviewers optionally provide to clarify their votes.</p>
	<p>Displays in tabular form the changes that were made to members upon submission of the change package.</p>
<b>Propagation Information</b>	<p>Displays a list of the change packages that were propagated by this change package and/or a list of the change packages that propagated this change package.</p>

To display or not display uncommitted entries (deferred entries and lock entries) in the GUI, select **View > Show Uncommitted**. Repeat to enable or disable the option.

To display or not display entries that are pending in the GUI, select **View > Show Pending**. Repeat to enable or disable the option.

To display or not display committed entries in the GUI, select **View > Show Committed**. Repeat to enable or disable the option.

# Editing a Change Package

You can update the summary, state, and description of a change package when necessary.

---

**NOTE** You can only edit a change package that you created, unless you have administrator permissions.

---

If reviews are mandatory, change packages in state **CommitFailed**, **Rejected**, or **Discarded** can be moved to the **Open** state as part of the state workflow.

Interface	Procedure
GUI	Select the change package you want to edit and select <b>Change Package &gt; Edit</b> .
Web	From a <b>Project</b> view, select the member that is associated with the change package you want to edit. Under the <b>C.P.ID</b> column, click the number link for the change package.

# Moving ChangePackage Entries

You can move change package entries from one change package to another. For example, if you checked in a member with the incorrect change package, you can move that change package entry to the correct change package, thereby ensuring auditing accuracy.

## **Key Considerations**

- Only the change package creator can move entries from one change package to another.
- Both the originating and the target change packages must be open.
- A change package administrator can move entries from one open change package to another open change package.
- You can only move entries from a change package in a `Closed` state if all of the following conditions are met:
  - You are the change package administrator
  - Reviews are not mandatory for the change package
  - The change package has not been propagated by another change package
- Moving a change package entry for a subproject operation moves any nested entries (member or subproject) at the same time.

## **To move change package entries in the GUI**

- 1 From the **Entries** panel on the **Change Package** view, select the change package entries you want to move.
- 2 Select **Change Package > Entry > Move**. The **Move Change Package Entry Settings** dialog box displays.
- 3 From the **Target Change Package** list, select the change package you want to move the entries to. The change packages are listed with the C.P. ID and the summary.
- 4 If you want to create a new change package for the entries, click **Create**.
- 5 If you are a change package administrator, the **Show all open Change Packages** option is available. Select the option to make change packages that you did not create available in the **Target Change Package** list.
- 6 When you are finished, click **OK**. The **Confirm Move Change Package** dialog box displays.
- 7 To move each entry presented, click **Yes**. To move all entries without viewing individual prompts, click **Yes to All**. The change package entries are moved to the selected change package.

**TIP** You can move change package entries by dragging them from one **Change Package** view to another.

## **To move change package entries in the Web interface**

- 1 From a **Project** or **Member History** view, click the change package ID. The **Change Package** view displays.
- 2 Click the **Entries** tab. The **Entries** panel displays.
- 3 Select the change package entries you want to move.

- 
- 4** Select **Actions > Move Change Package Entry**. The **Move Change Package Entry** dialog box displays.
  - 5** From the **Target Change Package** list, select the change package you want to move the entries to. The change packages are listed with the C.P. ID and the summary, for example, `10:1 help dialog broken`.
  - 6** If you want to create a new change package for the entries, click **Create**.
  - 7** When you are finished, click **OK**. The **Confirm Move Change Package** dialog box displays.
  - 8** To move the entry, click **Yes**. Repeat for additional entries. The change package entries are moved to the selected change package.

# Viewing a Generic Change Package

By default, MKS Integrity includes two change package types: MKS Integrity and MKS Implementer. A change package type consists of change package attributes and change package entry attributes. For specific information on MKS Implementer change packages, see the *MKS Implementer User Guide*.

Your administrator can create a custom change package type known as a *generic change package*. The ability to create a new change package type and modify the attributes is intended for use in creating custom integrations. For specific information on the generic change package type(s) used in your organization, see your administrator.

If you use generic change packages in your organization, you can view the details of a specific generic change package in the GUI and Web interface. For specific information on the generic change package type(s) used in your organization, see your administrator.

To display the **Generic Change Package View**, do one of the following:

- In the GUI, select a change package on an item's **Change Packages** panel, and click **Items > View Generic Change Package Details**.
- In the Web UI, click a change package ID link on an item's **Change Packages** panel.

**TIP** In the GUI, you can print the information summarized in the **Change Package View** by clicking **Print**.

## Generic Change Package Attributes Panel

Depending on your change package type, one or more of the following fields display.

Field	Description
<b>ID</b>	Change package ID number consisting of item ID and change package number, for example, 6:2.
<b>Type</b>	Type of change package, for example, MKS Integrity or Implementer.
<b>Status</b>	Change package state. If change package is in progress, it is <b>Open</b> ; if change package is completed, it is <b>Closed</b> .
<b>Summary</b>	Summary of change package.
<b>Created By</b>	Who created change package and when.
<b>Created Date</b>	Date change package was created.
<b>Entry Count</b>	Total number of members in change package.
<b>Server Hostname</b>	Name of MKS Integrity Server where change package resides.
<b>Server Port</b>	Port number on MKS Integrity Server where change package resides.
<b>Description</b>	Description of change package, if one was entered.
<b>Closed Date</b>	Date change package was closed.
<b>Subtype</b>	Type of change package: Development, Propagation, Staging, or Deployment. For more information on staging and deployment change packages, see <i>MKS Deploy 2009 Administration Guide</i> .
<b>Staging System</b>	Name of staging system associated with change package. For more information on staging systems, see <i>MKS Deploy 2009 Administration Guide</i> .
<b>Stage</b>	Stage associated with change package. For more information on stages, see <i>MKS Deploy 2009 Administration Guide</i> .

Field	Description
<b>Deploy Target</b>	Name of deploy target associated with change package. For more information on deploy targets, see <i>MKS Deploy 2009 Administration Guide</i> .
<b>Propagated</b>	Change packages change package was propagated by using the Apply CP command.
<b>Propagated By</b>	Change packages change package was propagated by using the Apply CP command.
<b>Deploy Request State</b>	Deploy request state associated with change package. For more information on deploy request states, see <i>MKS Deploy 2009 Administration Guide</i> .
<b>Deploy Request ID</b>	ID of deploy request associated with change package. For more information on deploy request IDs, see <i>MKS Deploy 2009 Administration Guide</i> .

## Generic Change Package Entries Panel

Depending on your change package type, one or more of the following fields display.

Field	Description
<b>Modifier</b>	State of change package entry either Committed or Pending. In the GUI, the field is hidden by default.
<b>Type</b>	Type of activity for change package entry.
<b>Member</b>	Individual member contained in change package.
<b>Revision</b>	Member's revision number.
<b>Project</b>	MKS Integrity project member belongs to, for example, c:/master_project/SourceCode/project.pj
<b>Variant</b>	MKS Integrity project's development path (if there is one).
<b>Date Changed</b>	Date change command occurred for listed member.
<b>Archive</b>	File location of MKS Integrity archive on MKS Integrity Server.
<b>Hostname</b>	MKS Integrity Server where change package resides, for example, xyz-server. In Web interface, clicking hostname hyperlink opens change package in MKS Integrity Web interface where you can work with change package's associated project and members. In GUI, field is hidden by default.
<b>Port</b>	Port on MKS Integrity Server where change package resides, for example, 7001. In GUI, field is hidden by default.
<b>Location</b>	Location of specific backing archive for member or location of backing project for subproject.
<b>Member Type</b>	Project element affected by change package operation. Available options are Subproject, Member, or Unknown.
<b>Lines Added</b>	Specific number of lines added by member operation. Option used for text archives.
<b>Lines Deleted</b>	Specific number of lines deleted by member operation. Option used for text archives.
<b>Bytes Added</b>	Specific number of bytes added by member operation. Option used for binary archives.
<b>Bytes Deleted</b>	Specific number of bytes deleted by member operation. Option used for binary archives.
<b>Text</b>	Whether change package entry contains text or binary archives.

---

# Submitting Change Packages

---

**CLI EQUIVALENT** `si submitcp`

---

You submit a change package in order to commit the associated changes to the repository. When reviews are mandatory, submitting a change package begins a review process that must be completed before the changes are committed to the repository.

You can submit change packages containing any combination of deferred, pending, and committed entries.

To submit a change package in the GUI, select the change package, and select **Change Package > Submit**.

## **Key Considerations**

- If exclusive locking is enabled, you cannot successfully submit a change package that contains a deferred check in if you do not have a lock on the member. You must check out the member before submitting the change package.
- Locked change package entries must be converted to deferred check in entries in order for the Submit CP operation to complete. The Check In dialog box displays when a locked entry is encountered. If the Check In operation results in a revision conflict (the revision being checked in is not the member revision) or a lock conflict (another user has an exclusive lock on the member), you can still proceed with the Submit CP operation by selecting the **Force Creation of New Branch** option.
- You cannot successfully submit a change package that contains deferred `Rename` or deferred `Move` entries if another user has a lock on the member.
- You cannot successfully submit a change package if the revision being updated is frozen.
- If a change package is submitted only containing pending entries, you are prompted to commit those pending entries, even if there are no deferred or lock entries.

---

# Closing a Change Package

---

**CLI EQUIVALENT** `si closecp`

---

Once the necessary changes are made to the members and subprojects, and they are committed to the repository, you can close the change package by selecting **Change Package > Close**.

In addition to this procedure, you can close a change package in the GUI by selecting the **Close Change Package** option when performing member or subproject operations.

If workflows are enabled, closing a change package allows the item to move forward in the MKS Integrity workflow.

---

**NOTE** You can only close a change package that you have created unless you have administrator permissions.

---

If reviews are mandatory, you cannot close a change package. It must follow the review process and be closed by the MKS Integrity Server.

# Discarding Change Packages

---

**CLI EQUIVALENT** `si discardcp`

---

MKS Integrity provides a way to remove change packages from active use by discarding them when they are not needed.

Interface	Procedure
GUI	Select the change package you want to edit and select <b>Change Package &gt; Discard</b> .
Web	Display the change package you want to discard in the <b>Change Package</b> view. Select <b>Actions &gt; Discard Change Package</b> .

Change packages can only be discarded if they are in the `Open` or `Rejected` states. To discard a change package, that change package must be both created by you and not contain any committed entries. However, a change package administrator may discard a change package created by another user.

---

**TIP** You can remove specific change package entries if required using the `discardcp` command.

---

When a change package is discarded, the following happens where applicable:

- All uncommitted entries are removed from the change package.
- All deferred operations corresponding to deferred entries are reverted.
- Locks on members corresponding to lock entries are released.
- All pending operations corresponding to pending entries are reverted and appear as discarded entries in the review log (to preserve the review history).
- All pending revisions that correspond to pending entries are deleted.
- Any archives created for pending members associated with pending entries in the change package are deleted from the server (pre-existing archives that were shared to create a pending member are not deleted).

## **Key Considerations**

- The change package ID for the discarded change package can never be used for any future change packages that are created.
- Discarded change packages have a state of `Discarded` and can still be viewed using **Change Package > Find** and filtering by the Committed or Pending category.
- If the change package has undergone a review, the review log persists.
- You can open and reuse discarded change packages.

---

# Change Package Reviews Overview

A change package review is a review of changes by specified reviewers before the changes are committed to the server repository.

If change package reviews are mandatory globally, then all change packages must progress through the change package review process.

If change package reviews are mandatory at the project level only, then a change package only progresses through the review process if it contains at least one entry associated with a member in a project that requires a review. Change packages follow the review process before the changes are successfully committed to the server repository. All other change packages function as non-review change packages.

A *change package reviewer* is a user specified by your administrator to review change packages containing members associated with specific projects. The reviewer may be individually specified or a member of a specified reviewer group. In the case of a reviewer group, any member of that group casts an accept or reject vote on behalf of the entire group.

A *change package watcher* is a user specified by your administrator who is notified when a reviewed change package is closed after being successfully committed to the repository. Change package watchers may be individually specified or a member of a specified watcher group.

A *super reviewer* is a user with permission to vote on change packages, but who is not required to be a listed reviewer for the change package. Voting as a super reviewer overrides all other votes. For example, casting an accept vote as a super reviewer is sufficient for accepting the change package.

## How Change Package Review Works

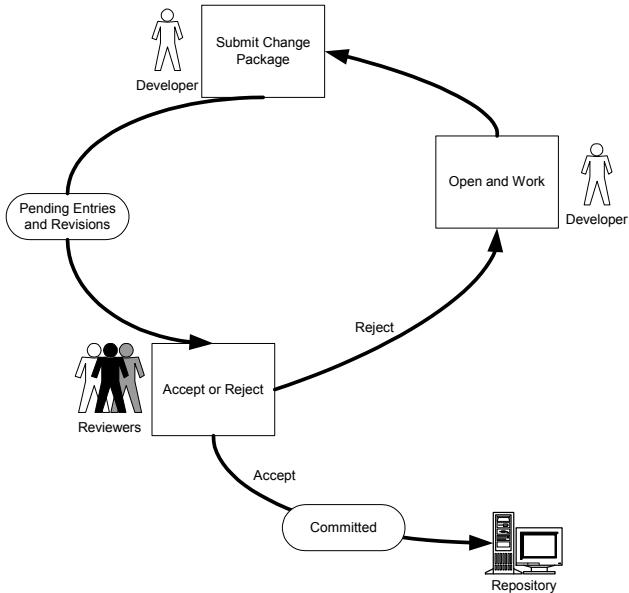
The following summarizes how the change package review process works:

- 1 The review starts when a developer (change package creator) submits a change package containing deferred and member lock entries and subproject entries (or any combination of deferred, pending, or committed entries).
- 2 MKS Integrity creates a pending change package entry for each deferred and member lock entry, and, where necessary, pending revisions.
- 3 Change package reviewers (possibly a mentor or a senior developer) accept or reject the change package.
- 4 The change package entries are either committed to the database (accepted case), or the developer opens the change package and continues development (rejected case).

---

**NOTE** Although the review process describes submitting change packages with deferred entries thereby creating pending entries, if deferred operations are not mandatory you can create pending entries at the time the operation is completed by clearing the deferred option. You can then submit the change package containing the pending entries for review. Subproject operations are always created as pending entries in a change package when reviews are mandatory.

---



*Change Package review process*

## ChangePackage Review Benefits

The benefits of using reviews are:

- Reviews provide a formal and enforceable review process.
- Reviews force changes to be reviewed before they are committed to the repository thereby providing control over what changes are accepted into a project by making operations pending. Unlike deferred operations, which are stored only client side, pending operations are stored server side and so are visible to all users. This can be useful near the end of a release cycle as a pre-commit review of changes before they are included in a release build.
- Pending revisions are not a part of the current state of the project until they have been reviewed (not at member revision); however, they remain in the member history and can be checked out (without a lock) by users (other than the creator) for review.
- If the project is one that users will be building from, reviews can remove the need for using a variant project to review changes manually.
- Reviews provide an alternative to manual post-commit review while recording all of the review information.

## ChangePackage Review Workflow

A change package under review progresses through states in a workflow.

The following table provides details on change package states. Where specified, some are only used in the review workflow:

Change Package State	Details
<b>Open</b>	Only state where work can be performed using a change package (new entries created).
<b>Submitted</b>	State the change package is in while it is being reviewed. All operations are pending.

Change Package State	Details
<b>Accepted</b>	Intermediate state denoting that the change package is accepted by all of the reviewers. MKS Integrity automates the state change from <b>Accepted</b> to <b>Closed</b> if the changes are successfully committed to the repository.
<b>CommitFailed</b>	State signifying that the pending changes could not be committed to the repository.
<b>Rejected</b>	State denoting that the change package is rejected by a reviewer. Creator must manually move the change package to <b>Open</b> to continue development.
<b>Discarded</b>	Empty change packages or change packages with changes that do not need to be committed to the repository are moved by the creator to the <b>Discarded</b> state. Discarded change packages can be opened at a later date if needed.
<b>Closed</b>	End state for the change package when pending changes are successfully committed to the repository.

The following is the change package progression through the workflow:

- 1 A change package is created and in the **Open** state. The developer adds entries to the change package.
- 2 The developer submits the change package to begin the change package review process, and MKS Integrity moves the change package to the **Submitted** state. An e-mail automatically notifies the reviewers of the change package submission (if the server is configured to send e-mail notifications). The e-mail contains both change package and review information.
- 3 The reviewer or reviewers, either accept the change package or reject it. The following can then happen:
  - If all individual reviewers and at least one reviewer from a reviewer group (if any exist) accept the change package, it is moved to the **Accepted** state. For each vote cast by a reviewer, MKS Integrity sends the reviewers an e-mail notification of the accept vote. When all reviewers have voted to accept the change package, MKS Integrity sends each reviewer and the creator an e-mail notification that the change package is accepted.

MKS Integrity then commits the changes to the repository, and then closes the change package. If MKS Integrity fails to commit the changes to the repository, it moves the change package to the **CommitFailed** state and an e-mail notification is sent to the creator stating the commit failure and the error associated with that failure.

From the **CommitFailed** state, once the commit failure reason is remedied, the creator can submit the change package again. Since the review is already completed, the change package moves to the **Closed** state without requiring passage through the review process an additional time, or any additional e-mail notifications to be sent.

From the **CommitFailed** state, the creator can also move the change package back to **Open**, continue development, and then submit it for review again.

---

If an accepted change package is successfully committed to the repository, MKS Integrity then closes the change package and an e-mail notification is sent to the change package watchers. Change packages in the `Closed` state cannot be opened.

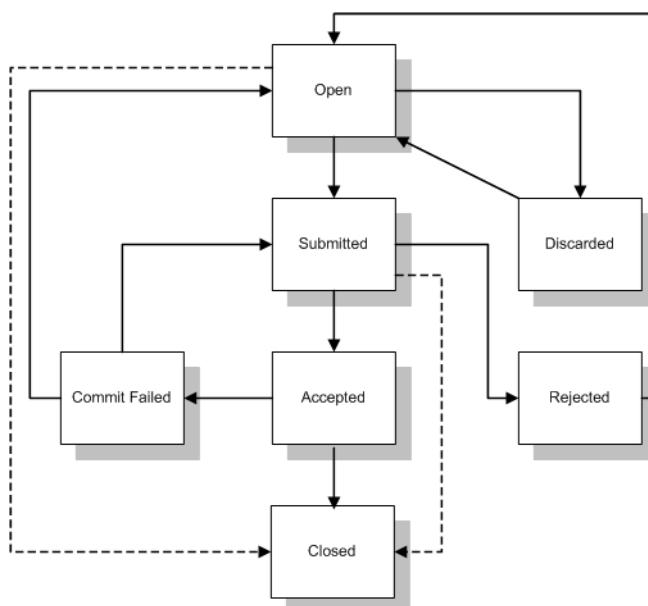
- If a reviewer (either an individual or a group member) rejects the change package, MKS Integrity moves it to the `Rejected` state and an e-mail notification is sent to each reviewer and the creator. The creator of the change package then moves the change package to the `Open` state (by editing the change package and changing the state), continues development, and then submits the change package again.

---

**NOTE** The creator of the change package continues development on pending revisions by checking them out.

---

The user can also discard the change package if it is no longer needed for development (thereby discarding entries contained in the change package). Change packages in the `Discarded` state can be moved back to the `Open` state if they are needed again.



*Change Package State Workflow*

## Change Package Entry Categories

There are four categories of change package entries: deferred, pending, discarded, and committed.

The following are change package entry type categories and their descriptions:

- deferred entries appear only when the change package is in the `Open` state. When the change package is submitted for review, deferred entries become pending entries. If exclusive locking is enabled, you cannot submit a change package that contains deferred check in entries that do not have corresponding locks.
- pending entries can appear when a change package is in the `Open`, `Submitted`, `Accepted`, `Rejected`, and `CommitFailed` states..
- discarded entries can only be viewed from the review log.

- 
- committed entries are entries that are committed to the repository. They are denoted simply by the change package entry type name, for example, `Update`.

---

**NOTE** For change packages that are not under review, only the deferred and committed entry types are available for use in MKS Integrity.

---

## Pending Operations in Change Packages

*Pending operations* are operations whose proposed changes reside on the server, but have not been committed to the repository.

The changes are committed to the repository once the change package is accepted upon review. Pending operations can create pending revisions, members, and subprojects.

---

**NOTE** Pending revisions are not currently created for subproject operations.

---

When reviews are not mandatory, pending entries are created when the changes are not successfully committed to the repository. MKS Integrity moves the change package to the **CommitFailed** state. You can open the change package and then submit it again, or continue development and submit it later.

Pending operations cannot be reverted, but must be discarded from the associated change package.

A *pending member / subproject*, is a member or subproject that is associated with a pending operation that adds it to the project. The pending member or subproject is denoted by a pending icon in the **Name** column of the **Sandbox** and **Project** views.

# Reviewing Change Packages

If you administrator has set up e-mail notifications for change package reviews, e-mails are sent to change package reviewers, watchers, and the change package creator at points throughout the change package review process.

You can view all the change packages waiting for your review by selecting **Change Package > View My Reviews** in the GUI. The **My Change Package Reviews <user>** view displays.

**NOTE** It is possible for you to be a reviewer of a change package you created. Contact your administrator for more information.

You can use the **Resync CP** command to create working files in your Sandbox for all of the changes in the change package you intend to review. Once your review is complete, you can accept or reject the change package.

MKS Integrity provides review logs as part of a complete review audit process. A log consists of individual records for each reviewer. Each time the change package is submitted for review, a new log is created.

To view the change package review log in the GUI and Web, from the **Change Package** view click the **Review Log** tab. The **Review Log** panel displays.

## Accepting a Change Package

**CLI EQUIVALENT** `si acceptcp`

After a change package is submitted, each individual reviewer and one member of each reviewer group (if specified) in the reviewer list must accept the change package before it can be committed to the repository and then closed. To see a list of reviewers for a change package, view the change package review log.

**NOTE** To accept a change package, the MKS Integrity Client must be connected to the server on which the change package resides.

### To accept a change package in the GUI and Web interface

- 1 From the GUI, select the change package you want to accept and select **Change Package > Accept**, or in the Web interface, with a **Project** or **Member History** view open, click the change package ID. The **Change Package** view displays.

Select **Actions > Accept Change Package**. The **Accept Change Package** dialog box displays. For detailed information on the **Accept Change Package** dialog box, see “Accept / Reject Change Package Dialog Box” on page 424.

- 
- 2** To accept the change package, click **OK**. The accept vote is cast and recorded in the review log.

If there are reviewers in addition to yourself, an e-mail is sent to notify those reviewers that you have cast an accept vote for the change package.

When all of the individual reviewers and at least one reviewer from each reviewer group have accepted the change package, an e-mail is sent to notify the reviewers and the creator that the change package is accepted.

Once the change package is accepted, it is committed to the server repository. If the change package is successfully committed to the repository, it then moves to a state of **Closed**. For more information on change package states, see “ChangePackage Review Workflow” on page 418.

If the change package under review is closed, an e-mail notification of the **Closed** state is sent to the change package watchers.

All e-mail notifications contain change package and review information.

## Rejecting a Change Package

---

**CLI EQUIVALENT** `si rejectcp`

---

If a reviewer of a change package determines that invalid changes were made or additional development must be completed to resolve an item, the reviewer can reject the change package. A single reviewer rejection is enough to reject a change package, even if there are multiple reviewers. The creator of the change package can reject it without being listed as a reviewer. To view the list of reviewers and reviewer groups, view the change package review log.

### To reject a change package in the GUI and Web interface

- 1** From the GUI, select the change package you want to reject and select **Change Package > Reject**, or in the Web interface, with a **Project** or **Member History** view open, click the change package ID. The **Change Package** view displays.

Select **Actions > Reject Change Package**. The **Reject Change Package** dialog box displays. For detailed information on the **Reject Change Package** dialog box, see “Accept / Reject Change Package Dialog Box” on page 424.

- 2** To reject the change package, click **OK**. The change package moves to a state of **Rejected** and the vote is recorded in the review log. An e-mail notification of the change package’s rejection is sent to the reviewers and creator.

From the **Rejected** state, the change package can be discarded or reopened. For more information on change package states, see “ChangePackage Review Workflow” on page 418.

---

## Accept / Reject Change Package Dialog Box

Field	Description
<b>ID</b>	Change package ID number.
<b>Summary</b>	Summary description of change package.
<b>Creator</b>	User name of the user who created the change package.
<b>Review As</b>	User name of the reviewer accepting the change package.
<b>Capacity</b>	Capacity in which you are a reviewer. Due to factors determined by your administrator, the options in the list vary per user for each change package. Possible options are: <ul style="list-style-type: none"><li>■ Individual casts a vote as an individual reviewer in the reviewer list.</li><li>■ Member of &lt;group&gt; casts a vote on behalf of the entire group (only one user from a group is necessary to vote on behalf of the entire group).</li><li>■ All Reviewers casts a vote as a specific user and any group the reviewer belongs to.</li><li>■ Super Reviewer casts an overriding accept vote that is sufficient for accepting the change package even if there are additional reviewers. A super reviewer is not required to be a listed reviewer for the change package.</li></ul>
<b>Comments</b>	Comments on the change package elements.

---

# Reopening a ChangePackage

---

**CLI EQUIVALENT** `si opencp`

---

The Open state is the only state you can add new entries to a change package. As part of the review workflow, change packages in the following states can be reopened: CommitFailed, Rejected, and Discarded.

---

**CAUTION**

- Only a change package administrator can reopen a change package in the Closed state. Modifying closed change package contents can compromise the integrity of the server repository.
- A change package cannot be reopened if it has been propagated to another development path..

Interface	Procedure
GUI	Select the change package you want to reopen, and select <b>Change Package &gt; Reopen</b> .
Web	From a <b>Project or Member History</b> view, click the change package ID. From the <b>Change Package</b> view, select <b>Actions &gt; Reopen</b> .

---

# Apply CP Overview

---

**CLI COMMAND:** `si applycp`

---

You use the Apply CP command to propagate changes recorded in change packages from one project or development path to another. This enables you to propagate only the changes that you want to. The Apply CP command is most useful for building software.

The Apply CP operation reads the entries in a change package and updates the project to the revisions listed in that change package. This function of the command is an automated process of the Update Revision command. The Apply CP operation may also require that files or subprojects be added, dropped, renamed, or moved. This function of the command is an automated process of the member and subproject commands.

Apply CP only operates on closed change packages and cannot perform merging. You need to use the Resync CP command to merge files and resolve merge conflicts, tracking the changes and resolved conflicts in a propagation change package.

The Apply CP operation occurs in the project (whereas the Resync CP operation occurs only in the Sandbox). If you run the Apply CP command from a Sandbox, the Sandbox acts as a redirector to the project.

If you are applying a change package that contains moved members or subproject between two projects, MKS Integrity automatically updates both projects.

## Using Apply CP in Your Development Environment

This section provides an example to show how Apply CP can be used in your environment. In this example, the buildmaster brings a feature from the main trunk of project development and applies it to an earlier release.

The abcBusiness software company has released their Aurora software, version 3.0. When the release was completed, the project was checkpointed. The development team is now working on a new set of features for the next release, 4.0. A new feature for this release is a timestamp function. All the changes associated with the timestamp function are recorded in a set of change packages, or set of items, that isolates the feature from other features.

Now abcBusiness receives a request from a customer who has Release 3.0 but also needs the new timestamp feature for its global operations. The code in development for Aurora 4.0 is not stable enough for release and too many resources would be required to accelerate the release schedule. How can abcBusiness provide the timestamp feature without affecting the current release? Because the code for this feature is isolated within a set of change packages, the Apply CP command can be used to propagate the feature to the earlier, stable release.

However, without the functionality of Apply CP, the buildmaster at abcBusiness would have to search the required change package(s) manually and individually review all of the associated files to isolate the changes related to the feature. The buildmaster would then have to add, drop, rename, and move files manually; update file revisions; merge around unwanted revisions; merge in required changes; and merge out any unwanted changes.

Using the functionality of Apply CP, this complicated process becomes largely automated. In MKS Integrity, the Apply CP operation works directly in the project to add, drop, rename, and move files and subprojects, and update file revisions as required to create the desired change. MKS Integrity presents you with a list—the *backfill list*—including all change packages required to capture the

---

changes. In the Apply CP operation, you must either accept or decline the entire list—you cannot make selections. If you accept the list, the Apply CP command propagates the changes directly to the project. If you decline the list, the Apply CP command cannot complete.

If the Apply CP command fails because merging is required, you can run the Resync CP command. Resync CP works in your Sandbox and allows you to make selections from the backfill list. MKS Integrity then merges around unwanted changes and uses differencing to merge files.

The buildmaster at abcBusiness would:

- Create a variant project off the checkpoint for version 3.0. This variant project is isolated from the rest of the development team so that unwanted changes are not added to the main trunk of the development path.
- Use Apply CP to apply the change packages to the variant project. The change packages contain all the files that were changed or added to produce the timestamp feature. Apply CP is essentially adding the feature to the variant of Aurora 3.0.
- Create an executable of the software.

That executable can then be tested by Quality Assurance and shipped to the customer.

## Using Subprojects

Subprojects are created, added, moved, or dropped through the Apply CP command as a result of subproject operations included in a change package. Subproject operations are applied with the following restrictions:

- Add subproject operations are ignored if the subproject already exists in the target project and is configured the same way as is specified in the change package.
- You cannot apply configure subproject operations or operations that result in a reconfiguration of a subproject, for example, adding a subproject that already exists on a different development path.
- Move subproject operations are ignored if the subproject already exists in the target location.
- Drop subproject operations are ignored if the subproject does not exist in the target location.

If you cannot include all change packages with subproject operations in the Apply CP command, you have the option of applying subproject operations that are implied by member operations included in a change package. This is known as *implicitly* propagating subprojects, and is controlled by an option of the Apply CP command. Using implicit subproject propagation enables you to propagate subproject changes without needing to actually apply the change packages containing those subproject operations.

---

### NOTE

- Earlier versions of MKS Integrity (formerly Source Integrity or MKS Source) always used implicit propagation of subproject operations.
  - Using implicit subproject propagation can result in unsatisfactory results. For example, if a change package used to create a subproject was also used to add a member to the subproject, and you do not include this change package, you will get an incomplete subproject.
- 

### How Subproject Changes Are Applied Implicitly

If a member operation implies the creation or addition of a subproject, MKS Integrity tries to re-create the source environment's project tree in the target environment. If part of the source environment's

---

tree has been dropped, MKS Integrity attempts to re-create it based on the nearest appropriate subproject.

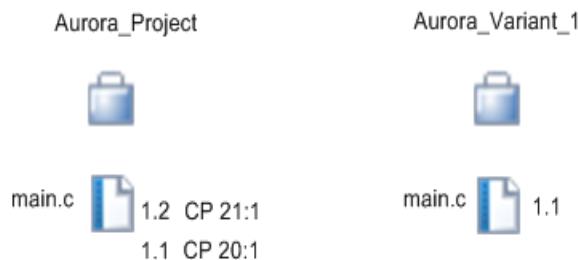
A subproject is added only if it is located under the project location for the target project environment and is configured the same way as the parent project. When added to a variant, the subproject is added as a variant subproject on the same development path as the parent project it is being added to. Apply CP does not detect or add any missing shared subprojects, even if they are present in the source project.

The creation or addition of subprojects takes place early in the Apply CP process. If you decide to cancel the operation when you are asked to confirm the changes to be applied, the subproject changes remain in the target environment.

## Applying a Change Package With No Dependencies

This example illustrates a simplified application of the Apply CP command in the main trunk of development for the Aurora project (`f:/Aurora_Project/project.pj`). The project member, `main.c`, includes a bug fix that allows the printing of version information. Item 21 addresses the bug fix and is associated with the file `main.c` (revision 1.2) through change package (CP) 21:1.

The buildmaster wants to pick up the changes that address the bug fix and apply these to a variant project, `Aurora_Variant_1_0`. In the variant project, `main.c` is at revision 1.1.



*Before applying a change package (simple case)*

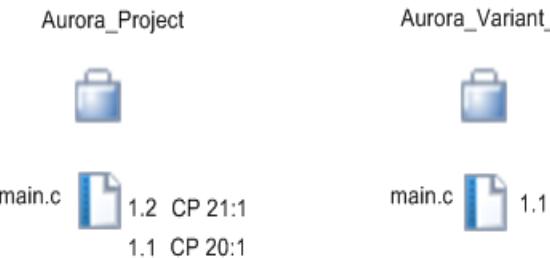
To get the bug fix for the variant project (`f:/Aurora_Variant_1_0/project.pj`), the buildmaster uses the `si applycp` command to apply CP 21:1:

```
si applycp -P f:/Aurora_Project/project.pj --devPath Aurora_Variant_1_0 21:1
```

The command runs as follows:

```
Applying change packages...
21:1
***The following set of operations will be performed:
Project: f:/Aurora_Project/project.pj[Aurora_Variant_1_0]
Member main.c: update member revision to Revision 1.2
Are you sure you wish to proceed? [yn]<n>: y
```

Because CP 21:1 included only an updating of `main.c` from revision 1.1 to revision 1.2, Apply CP updates the revision for `main.c` from 1.1 to 1.2 in the variant project.



*After applying a change package (simple case)*

## Using the Apply CP Backfill List

If the change packages you are applying are dependent on other change packages, the Apply CP command presents you with a backfill list that includes all of the required change packages. The following examples illustrate how the backfill list works in the Apply CP command.

In the Apply CP operation, you must accept the entire backfill list or the operation fails. If you do not want to accept the entire backfill list, you must instead perform a Resync CP operation. The Resync CP command allows you to merge around unwanted revisions in your Sandbox.

---

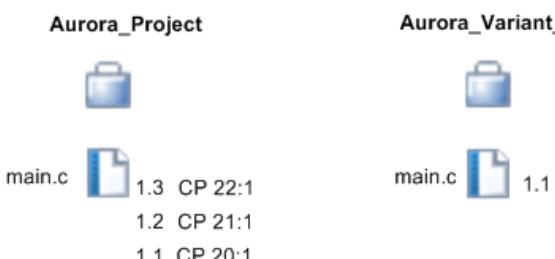
**NOTE** The backfill list is not used when applying subproject operations. You must explicitly specify all change packages containing subproject operations that you want to apply or select **Implicitly** for the **Subproject Propagation** option.

---

### Using the Backfill List to Include Required Previous Revisions

In this example, the main project, f:/Aurora\_Project/project.pj includes two bug fixes. Item 21 addresses the first bug fix and is associated with the file `main.c` (revision 1.2) through change package (CP) 21:1. Item 22 addresses the second bug fix and is associated with the file `main.c` (revision 1.3) through CP 22:1.

The buildmaster wants to pick up the changes that address the second bug fix and apply these to a variant project, `Aurora_Variant_1_0`. In the variant project, `main.c` is at revision 1.1.



*Before applying a change package that requires backfilling*

To pick up the bug fix for the variant project, the buildmaster uses the `si applycp` command to apply CP 22:1. By default, the backfill option is set to **Entire Change Packages** (`--backfill=cp`). The buildmaster enters the command:

```
si applycp -P f:/Aurora_Project/project.pj --devPath Aurora_Variant_1_0 22:1
```

The command runs as follows:

```
Applying change packages...
```

```
22:1
```

```
The following warnings have occurred:
```

```

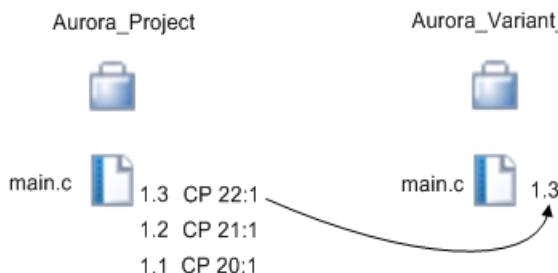
-----
The change package(s)
21:1 -- main.c(1.2)
are required in order to apply this list of change packages.
They will be automatically added to the list, since the backfill option is set to
Entire Change Package(cp).

-----
*** The following set of operations will be performed:
Project: f:/Aurora_Project/project.pj[Aurora_Variant_1_0]
Member main.c: update member revision to Revision 1.3
Are you sure you wish to proceed? [yn] (n): y

```

If you choose to proceed, you receive a notification listing the updates that have been processed, and any updates that were not processed.

Apply CP updates the revision for `main.c` from 1.1 to 1.3 in the variant project. Revision 1.2 is automatically added to the variant project because it was accepted as part of the backfill list (CP 21:1).

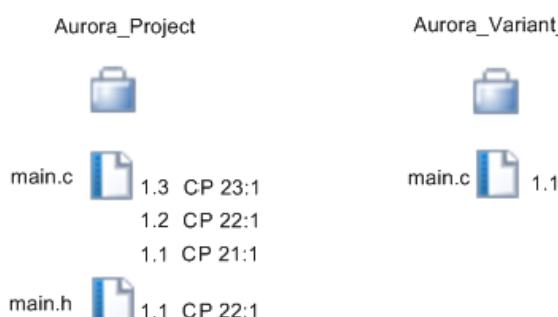


*After applying a change package that requires backfilling*

### **Using the Backfill List to Include a New File**

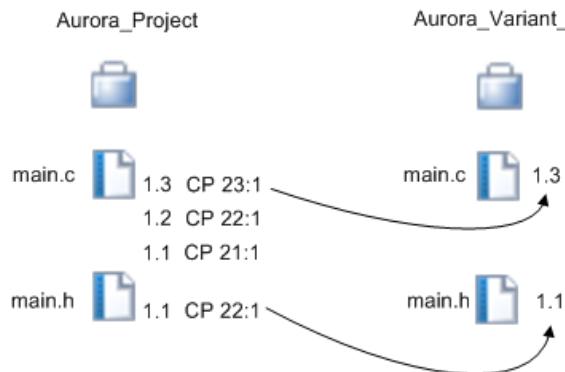
This example illustrates how Apply CP handles a more complex change package – one that contains code modifications that are dependent on a new file. In this example, `main.c` is revised to call a value defined in a new file, `main.h`.

The developer working on the code has checked in all these changes and associated both files with CP 22:1. Development work then continues to include a further revision to `main.c` which is checked in at revision 1.3 and associated with CP 23:1. The main project therefore contains `main.c` at revision 1.3 and `main.h` at revision 1.1.



*Before applying a change package that contains an associated file*

The buildmaster now wants to incorporate the changes into the variant project. The buildmaster therefore uses the Apply CP command to apply CP 23:1 to `Aurora_Variant_1_0`. This updates `main.c` to revision 1.3 and adds `main.h` revision 1.1.



*After applying a change package that contains an associated file*

### Using the Backfill List to Apply Back Revisions Only

The default backfill option for Apply CP is to propagate all entries required by the specified change package, including all entries in any change packages in the backfill list. However, there is a special case where you might want to backfill by **Back Revisions Only**. This option only updates the target environment based on the directly associated revisions. It does not process any change packages that are associated with intermediate revisions.

For example, if you chose the **Back Revisions Only** option in the previous example (“Using the Backfill List to Include a New File” on page 430) `main.c` would still be updated to 1.3, but `main.h` would not be propagated since it is an entry in a backfill change package.

This option is useful where you are propagating a file that collects information and therefore tends to be affected by all changes. In this case, you are only interested in the latest revision, not the underlying changes that occurred as a result of all the changes to related files.

## Using a Propagation Change Package With Apply CP

You can use another change package to record all the member and subproject changes made by an Apply CP operation. A change package used for this purpose is called a *propagation* change package.

**Note** If your administrator has set up change packages to be mandatory, you must specify a propagation change package.

Using a propagation change package makes it easier to track what changes have been propagated and to propagate the same changes to multiple variants. Any change packages that the applied change package is dependent on which have already been applied to the project by a previous Apply CP operation do not appear in the backfill list. You receive a warning message about the change packages that have already been applied.

You do not create a propagation change package – you create a normal change package and propagation information is recorded in the change package when you specify it during an Apply CP operation. For the greatest level of control in isolating changes, it is recommended that you start with an empty propagation change package. When the propagation change package contains no previous entries, the only additions will be those that specifically relate to the changes in question.

---

If reviews are mandatory, the changes made by an Apply CP operation are recorded as pending entries in the propagation change package. The change package must then be submitted, which starts the review process.

Once a propagation change package has been applied, you can see the list of change packages that it propagated in the **Change Package** view. For example, if you created propagation change package 10.6 to record changes made by an Apply CP operation, you would see the propagated change packages in the **Change Packages that this Change Package propagated** field on the **Propagation** tab of the **Change Package** view for 10.6. If you then applied propagation change package 10.6 to a different variant, and created propagation change package 10.8 to record the changes, you would see change package 10.8 listed in the **Change Packages that this Change Package was propagated by** field for 10.6.

## Resolving Conflicts

The Apply CP command detects as potential conflicts:

- files that have been dropped or moved, but would be re-added by Apply CP due to a revision update on that file in the set of change packages processed
- file additions due to a revision update in the set of change packages processed, where an alternate member using the same backing archive already exists

File additions due to an explicit request to add them from the set of change packages processed are *not* considered conflicts.

If merging is required, the Apply CP operation fails. You need to use the Resync CP command to merge files and resolve merge conflicts, tracking the changes and resolved conflicts in a propagation change package.

# Apply CP Procedure

This section describes the step-by-step procedures required to perform the Apply CP command in the GUI.

**CAUTION** You cannot undo an Apply CP operation. Therefore, before applying any change packages, you should checkpoint your project. You can then use the Restore Project command to revert to the earlier version of the project.

**NOTE** A helpful practice prior to using Apply CP, is to start with a Resync CP operation in a Sandbox, and then build and test the results, even if no merges are required. Because the operation may be creating a combination of source code that has never existed before, this step ensures that the results will build and work. Once you are certain of the results, you can then use the Apply CP command and work directly in the project.

## To apply a change package in the GUI

- 1 Select **Change Package > Apply**.
- 2 If you have a project selected, proceed to step 10. If you do not have a project selected, the **Apply Change Packages** wizard opens, displaying the Specify the Project panel.
- 3 Click **Select**. The first panel of the **Specify the mainline or variant project** wizard displays.
- 4 Enter a project name and proceed to step 6, or click **Select** to choose a project from the list.

**NOTE** If you are applying the change package(s) to a variant or build subproject, only enter the path and name of the root project in this field. You specify the subproject later in the procedure.

- 5 Click **Next**. The second panel of the **Specify the mainline or variant project** wizard displays.
- 6 Select the type of project you want to apply the CP(s) to by clicking a project type option. The available types are:
  - **Normal** applies the CP (s) to a project based upon the current state of the project. Click **Finish** and proceed to step 10.
  - **Variant** applies the CP(s) to a project based on a specific development path.

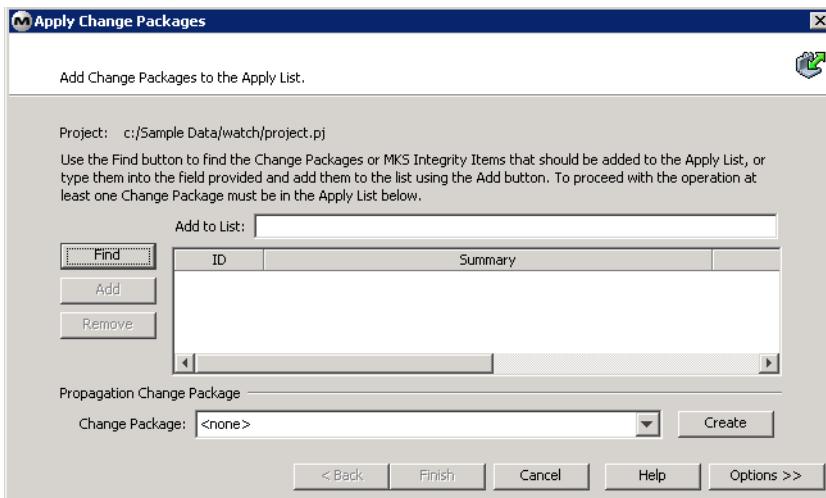
From the **Development Path Name** list, select a development path name, for example, *Aurora\_Beta\_Variant*.

**NOTE** The **Variant** option is unavailable if there are no available development paths.

- 7 If you are applying the CP(s) to a subproject, click **Next** to select a subproject. The third panel of the **Specify the mainline or variant project** displays.
- 8 Expand the project to select the specific subproject that you want to apply the CP(s) to. Click **Finish**.

**NOTE** There are rules that control what project configuration you can jump to. If your selection breaks any of the rules, you cannot apply the change package(s).

- 9 Click **Next**. The Apply List panel of the **Apply Change Packages** wizard displays.



- 10 To add change packages to the Apply List, click **Find**. The **Find Change Package** dialog box displays.  
If you know the ID number of the change package(s) or item(s) you want to add, you can also enter that number in the **Add to List** field and then click **Add**. For multiple numbers, include a space between each change package ID.
- 11 Select filter criteria for the change package, or if workflows are enabled, select a query.
- 12 After you specify the filter criteria or query, click **OK**. The **Select Change Package(s)** dialog box displays, populated with the filter or query results.

**NOTE** When using the Apply CP command, you can only apply closed change packages. Because you cannot apply an open change package to a project, the option for **Allow Open Change Packages** is disabled by default. If you need to work with an open change package, you must use the Resync CP command.

- For detailed information on the **Select Change Packages** dialog, see “Select Change Package(s) Dialog Box” on page 436.
- 13 To review details of a change package before applying it, right-click the change package and select **View Change Package Details**. The **Change Package** view displays for the selected change package.
  - 14 To add the change packages selected, click **OK**. The added change packages are displayed in the Apply List.  
To remove a change package from the Apply List, highlight the change package (or press CTRL and click to highlight multiple members), and then click **Remove**.
  - 15 To select the command options you want MKS Integrity to use when carrying out the Apply CP operation, click **Options**.  
For detailed information on the Apply CP options, see “Apply Change Packages Options: General” on page 436 and “Apply Change Package Options: Advanced” on page 437.
  - 16 To record changes to members as a result of the Apply CP operation in a propagation change package, select a change package in the **Change Package** field or create a new one. Only open change packages are available for selection.

- 
- 17 To apply the list of change packages to the project, click **Finish**. If additional change packages are required to apply the selected change package, the **Confirm Project Backfill** dialog box displays.

---

**NOTE** Any change packages that the applied change package is dependent on which have already been applied to the project by a previous Apply CP operation do not appear in the backfill list. For operations involving a large number of change packages, it may take several seconds for these calculations to be performed and the backfill dialog box to display. You receive a warning message about the change packages that have already been applied.

---

- 18 To accept the backfill list and have the listed change packages also applied to the project, click **Yes**.

You must accept the backfill list as presented or the Apply CP operation cannot be completed. If you do not want to accept the entire list, you must use the Resync CP command instead. The **Confirm Change Package Application** dialog box displays indicating the operations to be performed.

- 19 To complete the operation(s) and apply the selected change package(s), click **Yes**. The **Change Package Processing Complete** dialog box displays, listing the updates that have been processed, and any updates that were not processed.

Click **Save** to save the contents of the dialog box in an external text file.

---

**NOTE** To see any change packages that have been applied between two revisions of a project, use the **View Project Differences** command and, under **Preferences**, select the option for **Show Applied Change Packages**.

---

## Select Change Package(s) Dialog Box

Table Column	Description
C.P.I.D.	The identification number of the change package (in the form <code>x:y</code> , where <code>x</code> is the container ID (if MKS Integrity is enabled, it is also the item ID), and <code>y</code> is the change package index number.)
State	The state of the change package.
Summary	The description of the item associated with the change package.

## Apply Change Packages Options: General

Apply CP General Command Option	Function
<b>Confirm Actions</b>	Causes MKS Integrity to confirm all operations with you before proceeding with them.
<b>Notify When Complete</b>	Displays a detailed report when the operation is complete, listing the actions that were completed, along with any errors that were encountered. You can select from the following options: <b>Never</b> to never display the report. <b>On Completion</b> to always display the report. <b>On Error</b> to display the report only when errors were encountered.
<b>Create Variants</b>	Causes MKS Integrity to create new variant subprojects within the target variant project as required. This allows the subproject structure of the target variant to mirror the structure of the source project. If this option is not set, variant subprojects are not created as needed and any members in those subprojects are ignored (a warning displays).
<b>Close Change Package</b>	Closes the propagation change package used to record the changes.
<b>Backfill</b>	Determines how MKS Integrity treats historic revisions required by the specified change packages. For the Apply CP command, the backfill option is set to <b>Entire Change Packages</b> by default. You can select from the following options: <b>Entire Change Packages</b> chooses all historic revisions required by the specified change packages and applies them by updating member revisions, and adding, dropping, renaming, or moving files. The user is not prompted to confirm the backfill list. <b>Back Revisions Only</b> processes only the specified change package(s) and chooses only directly associated revisions. It does not process any change packages that are associated with intermediate revisions. <b>Error</b> terminates the operation if other change packages are required but are not specified. <b>Skip Revisions</b> causes MKS Integrity to merge around specified backfill revisions. Because the Apply CP command does not perform merging, this is treated as an error. <b>Ask to Specify</b> allows you to select the specific change packages you want to include. For the Apply CP operation, a list of additional change packages displays. The presented list of change packages cannot be manipulated. You must either accept the entire list or the operation fails.

## Apply Change Package Options: Advanced

Apply CP Advanced Command Option	Function
<b>Use Master</b>	<p>Causes MKS Integrity to operate on the top-level project if the target is a subproject. Changes are applied throughout the top-level project's hierarchy. If this option is not set, changes are only applied to the target project and its subprojects; other changes are ignored.</p> <p>When spanning variant projects with the <b>Use Master</b> option enabled, the variant of the top-level project associated with the spanned project is evaluated and the subproject, if missing in the variant, is added.</p>
<b>Span Projects</b>	<p>Applies all changes in the change package, even if this involves a different project than the one you initially targeted.</p> <p>If this option is not set, any changes for other projects are ignored.</p> <p><b>Caution:</b> This is the only command option that has the potential to affect other projects.</p>
<b>Already in Project is Error</b>	<p>Causes MKS Integrity to terminate the operation if the change being applied has already been applied to the project.</p> <p>If this option is not set, then the information displays as a warning.</p>
<b>Other Project is Error</b>	<p>Terminates the command if the change is for a project other than the one you initially targeted.</p> <p>If this option is not set, then the information displays as a warning.</p>
<b>Ignore Server in Change Package</b>	<p>Causes MKS Integrity to perform the Apply CP operation even if the current server does not match the server specified in the change package entries.</p> <p>If this option is not set, any mismatches are treated as errors.</p> <p>Note: This option is useful when a server is the same but is now identified differently, for example, the server name has changed. This option is required because projects are defined by their server and path. Use this option with caution.</p>
<b>Ignore Cross-Branch Entries</b>	<p>Causes MKS Integrity to use the most recent revision when members with revisions on multiple branches are being applied.</p> <p>If this option is not set, member revisions being applied from multiple branches are treated as errors.</p>
<b>Ignore Update Revision Entries</b>	<p>Ignores update revision entries in a change package. There is no user prompt. This option is to support compatibility with older versions of MKS Integrity.</p>
<b>Propagate Subprojects</b>	<p>Determines how MKS Integrity treats subproject operations required by the specified change packages.</p> <p>You can select from the following options:</p> <ul style="list-style-type: none"><li>■ <b>Explicitly</b> adds, drops, or moves a subproject only if there is an explicit command to do so in the change package.</li><li>■ <b>Implicitly</b> adds, drops or moves a subproject if the operation is implicitly required based on the change package entries. For example, if you are adding a member that is part of a subproject that does not exist in the larger project being updated, the subproject is added.</li></ul>

---

# Resync CP Overview

---

**CLI EQUIVALENT** `si resynccp`

---

You use the Resync CP command to preview the changes listed in change packages in the context of a Sandbox before propagating them to the project. The Resync CP command searches all files related to a selected change package, and all the change packages that may be associated with the related files. Resync CP is most useful for incorporating new software features or bug fixes during the software development process.

Resync CP provides additional options beyond those available for Apply CP, including options for allowing open change packages, merging, resolving conflicts, and continuing the resync operation when errors are encountered. MKS Integrity allows you to run the Apply CP command first and then, if required, run the Resync CP command to perform any required merge operations.

---

**NOTE** If you attempt to perform a Resync CP command on a member that was previously dropped from a project, MKS Integrity instructs you to manually re-add the member.

---

While working in your Sandbox, you can also use the Resync By CP command, which only processes the change packages associated with the member you are resynchronizing.

## Using Resync CP in Your Development Environment

This section provides an example to show how Resync CP can be used in your environment. The example also illustrates how Resync By CP can be used in a developer's Sandbox.

The development of a new release typically involves working on hundreds of files that span many MKS Integrity projects. Whenever a fix is made, or a feature added, it may require the manipulation of a single file, or many different files across multiple projects. A fix or new feature may also include new dependencies within the source code.

If developers resynchronize only a single file into their Sandboxes, their builds may break because of new dependencies in the code. Such broken builds cause delays and prevent the team from completing their work on time. However, it is possible to avoid this lost time by using the Resync CP command.

Resync CP allows developers to specify a change package and have all changes associated with that change package resynchronized into their Sandboxes. The commands save development time because they:

- automatically search for the required files
- determine what other change packages the selection is dependant on (this is known as the backfill list) and also resynchronize those change packages into the Sandbox

If the developer is working on a file conflict, Resync CP also merges new information into a file. The merge operation ensures that the Sandbox is up-to-date and that no changes are lost.

The Resync CP command also allows a developer to remove a bug fix or feature that is incomplete or not working.

---

## Using the Resync CP Backfill List

If the change packages you are applying are dependent on other change packages, the Resync CP command presents you with a backfill list that includes all of the required change packages. The following examples illustrate how the backfill list works in the Resync CP command.

---

**NOTE** The backfill list is not used when resyncing subproject operations. You must explicitly specify all change packages containing subproject operations that you want to resync or select Implicitly for the **Subproject Propagation** option.

---

### ***Using the Backfill List to Include Required Previous Revisions***

The following example illustrates a simplified application of the Resync CP command in the main trunk of development for the Aurora project (`Aurora_Project.pj`).

The project member `tool.c` includes a bug fix for item 24 and is associated with the file `tool.c` (revision 1.7) through change package (CP) 24:1.

The developer wants to pick up the changes that address the bug fix and apply these in a Sandbox. In the developer's Sandbox, `tool.c` is at revision 1.2.

To apply the bug fix in the Sandbox, the developer uses the `si resynccp` command to apply CP 24:1 as follows:

```
si resynccp -S c:/Aurora_Project/project.pj 24:1
Applying change packages...
24:1
```

The following warnings have occurred:

```
-----
```

The change package(s)

```
20:1 -- tool.c(1.3)
21:1 -- tool.c(1.4)
22:1 -- tool.c(1.5)
23:1 -- tool.c(1.6)
```

are required in order to apply this list of change packages. They will be automatically added to the list, since the backfill option is set to Entire Change Package (cp).

```
-----
```

\*\*\* The following set of operations will be performed:

```
Project: f:/Aurora_Project/project.pj
Sandbox: c:\Aurora_Sandbox\project.pj
```

```
Member tool.c: resynchronize to Revision 1.7
```

```
Are you sure you wish to proceed? [yn]<n>: y
```

In this case, the Resync CP command updates the working file revision for `tool.c` from 1.2 to 1.7 in the Sandbox. This is done by checking out `tool.c` at 1.7 into the Sandbox. The changes made from revisions 1.3 through 1.6 are already included in this checked out file.

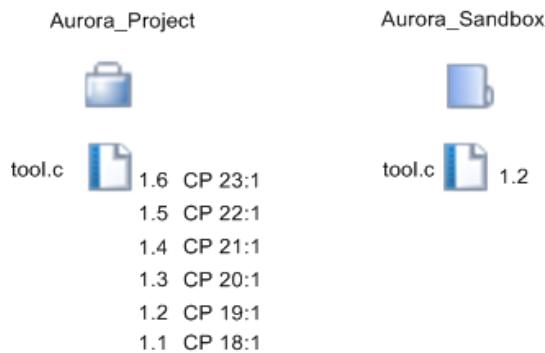
### ***Using the Backfill List to Merge Around Previous Revisions***

You can merge around some or all of the required change packages that are presented to you in the backfill list.

---

The main project, `Aurora_Project.pj`, now includes an additional bug fix for project member, `tool.c`. Item 23 addresses that bug fix and is associated with the file `tool.c` (revision 1.6) through CP 23:1.

The developer wants to pick up the changes that address the bug fix and apply these in a Sandbox. In the developer's Sandbox, `tool.c` is at revision 1.2.



*Picking up a specific change using the Resync CP command*

To pick up the bug fix, the developer uses the `si resynccp` command in a Sandbox. The developer wants to decide on the specific change packages to include in the operation, so he or she sets the backfill option to **Ask to Specify** (`--backfill=ask`). The command runs as follows:

```
si resynccp -S c:/Aurora_Sandbox/project.pj --backfill=ask 23:1
Applying change packages...
23:1
*** The following list of change packages are used by revisions before the
revision that you require. Each change package is given, along with the revisions
which require them:
20:1 -- tool.c(1.3)
21:1 -- tool.c(1.4)
22:1 -- tool.c(1.5)
Reply with:
y to pick up all these change packages, along with their associated changes,
s to skip all these revisions and merge around them (default)
c to cancel the command
or a space separated list of change package identifiers from the list given to be
*removed* from the list [y|s|c|#...]? 
```

---

**NOTE** When selecting the change packages from the backfill list in the command line interface, you enter the numbers for the change packages you want to **exclude** from the resync operation; in the GUI, you select the change packages you want to **include** in the resync operation.

The developer decides to merge around all the intermediate change packages and selects `s` (skip). The command continues as follows:

```
The following warnings have occurred:
-----
The following members require a merge to be performed:
    tool.c
You have not specified a change package, so merged members will not be locked. 
```

```
-----  
*** The following set of operations will be performed:  
Project: f:/Aurora_Project/project.pj  
Sandbox: c:\Aurora_Sandbox\project.pj  
Member tool.c: merge around differences: picking up revisions 1.2 through 1.6,  
excluding revisions 1.3, 1.4, 1.5, by checking out Revision 1.2 into the  
working file, and merging in the differences between Revision 1.6 and Revision  
1.2 with the differences between Revision 1.5 and Revision 1.3.
```

Are you sure you wish to proceed? [yn] (n) : y

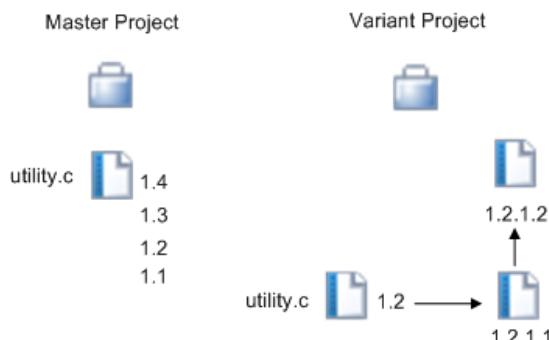
If you choose to proceed, you receive a notification listing the updates that have been processed, and any updates that were not processed.

Resync CP updates the working file revision for `main.c` by checking out revision 1.2 and then merging into the working file the differences between 1.2 and 1.6. The intermediate revisions are not added to the Sandbox because the skip option was selected. Because the resync operation does not use a propagation change package to record the changes, the merged member is not locked.

## Applying Changes From a Variant

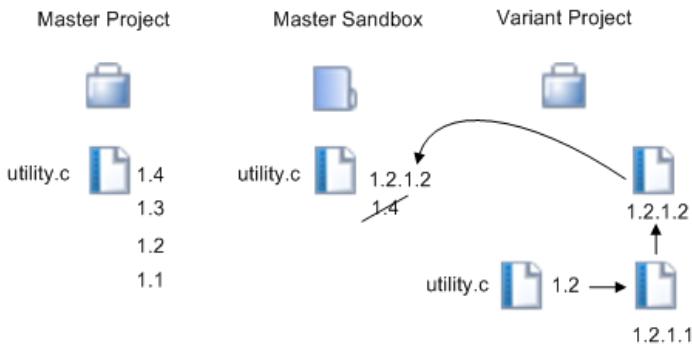
Working on a variant allows you to create a specific change, test it, and then bring that change back into the main trunk of development. Within reason, this can be done even when the main trunk includes further new development.

For example, a patch created in a variant project is needed in the master project. The file—`utility.c`—has a head revision of 1.4 in the project. In the variant project, the file `utility.c` version 1.2 is checked out to revision 1.2.1.1 and the code is revised. The file `utility.c` is then checked in at 1.2.1.2 and associated with CP 5:1.



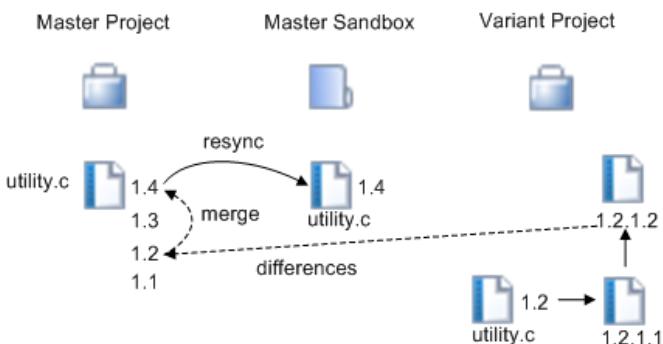
*Moving a patch from a variant project to a master project*

Moving the patch from the variant to the master requires a three-way merge operation using Resync CP. Because the master project contains further new development, updating the head revision of `utility.c` from 1.4 to 1.2.1.2 would cause the new development work in revisions 1.3 and 1.4 to be lost.



#### *The default Resync CP operation*

In this situation, you must use the **Merge On Branch** option (`--mergeOnBranch`). This option essentially allows the changes on the branch to be merged into the head revision file. Selecting **Merge On Branch** allows MKS Integrity to perform a differencing between revision 1.2 and 1.2.1.2 and then merge the result into revision 1.4. Once the Resync CP operation completes, the file must then be checked in to finalize the changes in the project.



#### *Using Resync CP with the Merge On Branch option*

#### **Cross-Branch Changes**

The **Ignore Cross-Branch Entries** option allows you to use the most recent revision when there are revisions of the same member on two different branches. This option can be used to accommodate the situation where you need to propagate changes from a variant that has temporary branches which were created in order to bypass locks. In this case, you do not want to include the changes on the branches, which have already been merged into the variant.

## Applying Change Packages From Two Variants

Another situation that warrants special consideration is the case where two change packages must be applied from different variants. Unless the correct procedure is used, Resync CP will fail because too many merges are required.

For example, the master project contains the file `patch.c` at revision 1.4. The variant project used by the buildmaster contains a fix (`patch.c`, revision 1.2.1.2, CP 9:1) that is being rolled out for a product patch. The variant project for the Maintenance Development team also includes a bug fix for `patch.c` at revision 1.3.1.2, and this revision is associated with CP 10:1. Both fixes are required for the master project.



*Performing a Resync CP operation on two change packages from different variants*

There are only two ways to successfully complete the preceding scenario without restrictions: address it manually, or perform Resync CP and Apply CP twice (once for each change package), checking in the merged changes at the head revision after each operation.

## Using a Propagation Change Package with Resync CP

You can use another change package to record all the member changes made by a Resync CP operation. A change package used for this purpose is called a *propagation* change package.

**NOTE** Even if your administrator has set up change packages to be mandatory, you do not need to specify a propagation change package for a Resync CP operation.

You do not create a propagation change package – you create a normal change package and propagation information is recorded in the change package when you specify it during a Resync CP operation. For the greatest level of control in isolating changes, it is recommended that you start with an empty propagation change package. When the propagation change package contains no previous entries, the only additions will be those that specifically relate to the changes in question.

The propagation change package is populated with all the member and subproject changes performed and the change packages propagated as a result of the resynchronization. You can discard undesirable member changes and add resolved merge conflicts to the propagation change package. You can also add entries to the propagation change package as required using MKS Integrity commands, for example, Check Out, Move, Rename, Update Member Revision, or Create Subproject.

Once all changes are completed, the propagation change package can then be submitted to update the project.

**NOTE** You can also submit the propagation change package without updating member revisions; then the Buildmaster applies the propagation change package at the time the software is built.

Entries in the propagation change package supersede corresponding entries in the resynchronized change packages. Thus, if you perform a merge from a branch and check the result into a propagation change package, the resulting revision supersedes any entries in the listed change packages that might be on branches.

### Advantages of Using a Propagation Change Package

Using a propagation change package ensures that other developers can apply the same set of change packages without duplicating the effort of running the Resync CP command and resolving any errors or merge conflicts.

Propagation change packages are also useful for assisting buildmasters to complete their work. When an Apply CP operation has failed, developers can use the Resync CP command on the same change

---

package, identify the dependencies and required merges, and include all the necessary changes in a single propagation package.

Propagation change packages allow you to incrementally propagate changes between development paths by repeatedly using the Resync CP command to collect project changes, rather than performing one large Resync CP operation that can take a long time.

If you use a propagation change package, any change packages that the applied change package is dependent on that have already been applied to the project through a previous Resync CP operation do not appear in the backfill list. You receive a warning message about the change packages that have already been applied.

It is important to note that while Resync CP can be used to apply a propagation change package, the results may not always be acceptable. For example, if your bug fix is in an existing project member, there would already be an archive for that member in the project. As a result, Resync CP would add the modified member on a branch. This additional branching might not be acceptable in your project.

### ***Example of Using a Propagation Change Package***

The abcBusiness company has two development teams:

- a Product Team that develops new features and software for the main release cycle
- a Maintenance Development Team that maintains the released software and addresses bugs that are identified by customers

The Product Team (PT) implements new features and designs on the main development path.

The Maintenance Development Team (MDT) works on a variant development path for Release 2.0 and fixes any problems in the newly-released product. The main goal of this team is to produce bug fixes for Release 2.0a. The work process for the MDT is:

- A bug is reported by a customer.
- A change package for the bug is created, in this case it has the container ID 1204. Workflows are enabled, so an item is created and then associated with a created change package.
- An MDT developer is assigned to fix the problem.
- The MDT developer creates a change package.
- The MDT developer makes the necessary changes and tests the code.
- The MDT developer checks the modified files back into the variant project, making sure to associate the files with change package 1204:1.

In this case, all the work of the MDT developer is now checked into the variant development path and will be part of release 2.0a. However, the MDT bug fix work needs to be transferred back to the main development path so that it can be incorporated into the next product release. A PT developer needs to pick up the changes that address the bug fix and apply them in a Sandbox. The Resync CP command is the best option for applying the new fix.

The PT developer creates a second change package, 1204:2 for the same item. The second change package includes the summary “Applied fix to main development path”. The PT developer starts the Resync CP command, selects the main development Sandbox, and the first change package—1204:1—in this item. The second change package—1204:2—is used as a propagation change package.

Once all merge conflicts have been resolved, the developer submits the propagation change package and MKS Integrity applies the changes from the referenced change packages.

The bug fix is now addressed in both the main and variant development paths, ensuring that the problem is fixed in the Release 2.0a and the next major product release.

---

## Resolving Binary Conflicts

When a conflict between two revisions of a binary file is encountered, MKS Integrity replaces the Sandbox working file for that member with the destination revision and marks it as a conflict. The destination revision displays in the propagation change package as a `Deferred Check In` entry (representing a deferred checkin of the working file). For example, if revision 1.1 and 1.1.1.1 are both update revision entries of a binary member in a change package that require a merge with member revision 1.2, then the contents of revision 1.1.1.1 are used for the Sandbox working file of member revision 1.2. Revision 1.2 displays as a `Deferred Check In` entry in the propagation change package. No comparison of binary file content is provided. When the change package is submitted (or member checked in) the member revision is 1.3.

You can resolve the conflict in one of the following ways:

- **Use Target Revision**

Submit the changes in the propagation change package, thereby checking in the destination revision for the binary member (or manually submit the deferred checkin of the destination revision).

- **Perform Manual Binary Merge**

Using a third-party tool outside of MKS Integrity, perform a binary merge on the affected member. Then submit the changes in the propagation change package (or manually submit the deferred checkin of the destination revision).

- **Use Original Revision**

Discard the `Deferred Check In` entry corresponding to the conflict, effectively using the original revision instead of the destination revision. To resolve conflict, indicate the desired revision by using a `Deferred Update Revision` operation that specifies the desired revision.

# Resync CP Procedure

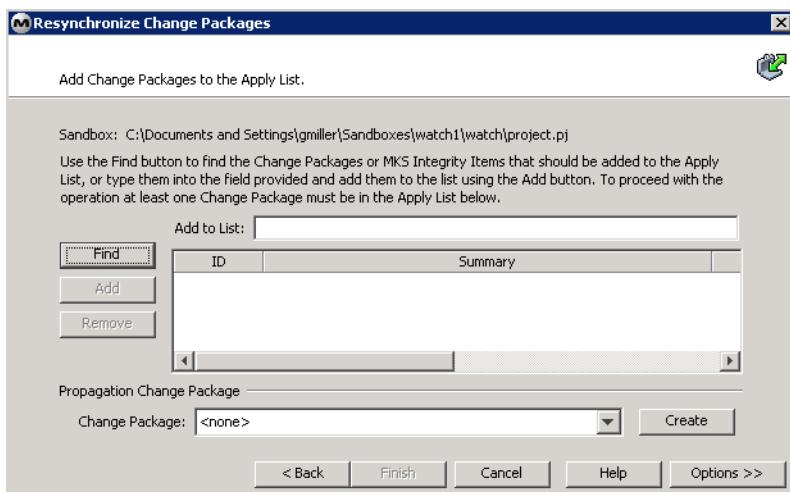
This section describes the step-by-step procedure required to perform the Resync CP command in the GUI.

Before performing the Resync CP command:

- Make sure all working files are in synch with the server
- Close all change packages you want to use for merging into the target environment

## To resynchronize a change package in the GUI

- 1 Select **Change Package > Resynchronize**. The **Select a Sandbox** dialog box displays.
- 2 Select the Sandbox you want the Resync CP command to operate on.
- 3 Click **OK**. The **Resynchronize Change Packages** wizard opens, displaying the **Apply List** panel.



- 4 To add change packages to the **Apply List**, click **Find**. The **Find Change Package Panel** displays. If you know the ID number of the change package(s) or item(s) you want to add, you can also enter that number in the **Apply to List** field and then click **Add**. For multiple numbers, include a space between each change package ID.
- 5 Select filter criteria for the change package, or if MKS Integrity is enabled, select a query. The **Select Change Package(s)** dialog box displays, populated with the filter or query results.

**NOTE** Open change packages are allowed when resynchronizing change packages. The option for **Allow Open Change Packages** is enabled by default. However, you cannot resynchronize open change packages if you are recording changes in a propagation change package.

- 6 To review details of a change package before including it in the resync operation, right-click the change package and select **View Change Package Details**. The **Change Package** view displays for the selected change package.
- 7 To add the change packages to the **Apply List** and return to the **Apply List** panel, click **OK**. The list of change packages displays in the **Apply List** panel.

To remove a change package from the **Apply List**, highlight the change package(s) and click **Remove**.

- 
- 8** To record changes to members as a result of the Resync CP operation in a propagation change package, select a change package in the **Change Package** field or create a new one. Only open change packages can be selected.
  - 9** To select the command options you want MKS Integrity to use when carrying out the Resync CP operation click **Options**. For detailed information on the Resync CP options, see “Resynchronize Change Packages Options: General” on page 448 and “Resync CP Options: Advanced” on page 450.
  - 10** To run the Resync CP command with the selected options, click **Finish**. If you selected the **Backfill** option **Ask to Specify**, the **Select Change Package(s) for Backfill** dialog box displays.

---

**NOTE** If you specified a propagation CP, any change packages that the applied change package is dependent on which have already been applied to the project by a previous Resync CP operation do not appear in the backfill list. You receive a warning message about the change packages that have already been applied.

---

- 11** To review details of a change package before including it in the resync operation, right-click the change package and select **View Change Package Details**. The **Change Package** view displays for the selected change package.
- 12** Select the change package(s) you want to include in the resync operation and click **OK**. The **Confirm Change Package Application** dialog box displays, providing information on the operations to be performed, and any warnings.
- 13** To complete the Resync CP operation, click **Yes**.

Depending on the preferences you have set for the Resynchronize command, you may be prompted to confirm overwriting your working files for the members being updated. If a merge is required, the Resynchronize command preferences also determines the type of merge used and the actions MKS Integrity takes in the event of a merge conflict.

- 14** MKS Integrity completes the Resync CP operation and the **Change Package Processing Complete** dialog box displays, listing the updates that have been processed, and any updates that were not processed.

To save the contents of the dialog box to an external text file, click **Save**.

---

**NOTE** To see any change packages that have been applied between two revisions of a project, use the **View Project Differences** command and, under **Preferences**, select the option for **Show Applied Change Packages**.

---

## Resynchronize Change Packages Options: General

Resync CP General Command Option	Function
<b>Confirm Actions</b>	Causes MKS Integrity to confirm all operations with you before proceeding with them.
<b>Notify When Complete</b>	Displays a detailed report when the operation is complete, listing the actions that were completed, along with any errors that were encountered. You can select from the following options: <b>Never</b> to never display the report. <b>On Completion</b> to always display the report. <b>On Error</b> to display the report only when errors were encountered.
<b>Create Variants</b>	Causes MKS Integrity to create new variant subprojects within the target variant project as required. This allows the subproject structure of the target variant to mirror the structure of the source project. If this option is not set, variant subprojects are not created as needed and any members in those subprojects are ignored (a warning displays).
<b>Continue on Errors</b>	Causes MKS Integrity to continue to process the change package if errors occur while resynchronizing. If you specify the <b>Notify When Complete</b> option, any errors are reported when the command is complete.
<b>Merge On Branch</b>	Causes MKS Integrity to perform a merge if the target revision is on a branch. MKS Integrity differences the two file revisions and merges any changes into the working file without modifying its revision number. You must then check in the working file to advance the revision to the next available revision number.
<b>Merge Type</b>	Determines what occurs if a merge is required to resynchronize the change package. <ul style="list-style-type: none"> <li>■ <b>Confirm</b> causes MKS Integrity to confirm the merge.</li> <li>■ <b>Cancel</b> causes MKS Integrity to stop the merge from occurring.</li> <li>■ <b>Automatic</b> causes MKS Integrity to perform the merge without prior confirmation.</li> <li>■ <b>Manual (Launch Tool)</b> launches your default merge tool.</li> </ul>
<b>On Conflicts</b>	Determines how merge conflicts are treated. <ul style="list-style-type: none"> <li>■ <b>Confirm</b> causes MKS Integrity to request input from the user to resolve the conflict.</li> <li>■ <b>Cancel</b> causes MKS Integrity to stop the resynchronize.</li> <li>■ <b>Mark For Later Merge</b> marks the conflicting lines so they can be addressed at a later date.</li> <li>■ <b>Launch Tool</b> causes MKS Integrity to launch your default merge tool.</li> <li>■ <b>Highlight Output File</b> highlights each line conflict in working file.</li> <li>■ <b>Error</b> causes MKS Integrity to display an error message.</li> </ul>
<b>Backfill</b>	Determines how MKS Integrity treats historic revisions required by the specified change packages. For the Resync CP command, the backfill option is set to <b>Entire Change Packages</b> by default. You can select from the following options: <ul style="list-style-type: none"> <li>■ <b>Entire Change Packages</b> chooses all historic revisions required by the specified change packages and applies them by updating member revisions, adding files, or dropping files. The user is not prompted to confirm the backfill list.</li> <li>■ <b>Back Revisions Only</b> processes only the specified change package(s) and chooses only directly associated revisions. It does not process any change packages that are associated with intermediate revisions.</li> <li>■ <b>Error</b> terminates the operation if other change packages are required but are not specified.</li> <li>■ <b>Skip Revisions</b> causes MKS Integrity to merge around specified backfill revisions.</li> <li>■ <b>Ask to Specify</b> allows you to select the specific change packages you want to include.</li> </ul>

---

Resync CP General Command Option	Function
<b>Already in Project is Error</b>	Causes MKS Integrity to terminate the operation if the change being applied has already been applied to the project. If this option is not set, then the information displays as a warning.
<b>Other Project is Error</b>	Terminates the command if the change is for a project other than the one you initially targeted. If this option is not set, then the information displays as a warning.

## Resync CP Options: Advanced

Resync CP Advanced Command Option	Function
<b>Use Master</b>	Causes MKS Integrity to operate on the top-level Sandbox if the target is a subproject. Changes are applied throughout the top-level Sandbox's hierarchy. If this option is not set, changes are only applied to the target Sandbox and its sub Sandboxes; other changes are ignored. When spanning variant projects with the <b>Use Master</b> option enabled, the variant of the top-level project associated with the spanned project is evaluated and the subproject, if missing in the variant, is added.
<b>Span Projects</b>	Applies all changes in the change package, even if this involves a different project than the one you initially targeted. In a Resync CP operation, this option allows MKS Integrity to search across local Sandboxes for all the entries contained in the selected change package(s). If this option is not set, any changes for other projects are ignored.
<b>Perform Merges</b>	Enable merging in the Sandbox if required.
<b>Allow Open Change Packages</b>	Allows MKS Integrity to work with open change packages.
<b>Already in Project is Error</b>	Causes MKS Integrity to terminate the operation if the change being applied has already been applied to the project. If this option is not set, then the information displays as a warning.
<b>Other Project is Error</b>	Terminates the command if the change is for a project other than the one you initially targeted. If this option is not set, then the information displays as a warning.
<b>Ignore Server in Change Package</b>	Causes MKS Integrity to perform the Resync CP operation even if the current server does not match the server specified in the change package entries. If this option is not set, any mismatches are treated as errors. Note: You must select the <b>Other Project is Error</b> option for the <b>Ignore Server in Change Package</b> option to work. This additional setting is required because projects are defined by their server and path.
<b>Ignore Cross-Branch Entries</b>	Causes MKS Integrity to use the most recent revision when members with revisions on multiple branches are being applied. If this option is not set, member revisions being applied from multiple branches are treated as errors.
<b>Ignore Update Revision Entries</b>	Ignores update revision entries in a change package. There is no user prompt. This option is to support compatibility with older versions of MKS Integrity.
<b>Propagate Subprojects</b>	Determines how MKS Integrity treats subproject operations required by the specified change packages. You can select from the following options: <ul style="list-style-type: none"><li>■ <b>Explicitly</b> adds, drops, or moves a subproject only if there is an explicit command to do so in the change package.</li><li>■ <b>Implicitly</b> adds, drops or moves a subproject if the operation is implicitly required based on the change package entries. For example, if you are adding a member that is part of a subproject that does not exist in the larger project being updated, the subproject is added.</li></ul>

---

# Apply CP and Resync CP Overview

Apply Change Package (Apply CP) and Resynchronize Change Package (Resync CP) represent two of the most powerful MKS Integrity features. In an environment where development is constantly evolving to include bug fixes or new features, Apply CP and Resync CP allow you to identify and incorporate only the specific bug fixes or content that you want to include in a new project. The functionality allows you to move specific changes—whether from the master project to a variant, from a variant to the master project, or from a variant to another variant.

Variant projects are projects that branch from the main trunk of development. Variant projects are identified through development paths.

Apply CP and Resync CP rely on the use of change packages to track individual changes that modify project content or create new content.

If a development team does not use the change package methodology, isolating specific content becomes a complex, manual task. In a large code project, this could mean searching hundreds of files to determine which ones are related to a specific item. To build the project, it would then be necessary to add, drop, rename, and move files; update file revisions; merge around unwanted revisions; merge in required changes; and merge out any unwanted changes.

If a development team uses change packages consistently, MKS Integrity can isolate all changes related to a specific item because this information is recorded as part of the change package. Once the dependencies are calculated, Apply CP performs the operations required to propagate the desired changes. If merging is required, you can use the Resync CP command. Resync CP allows you to either merge in desired changes or merge around unwanted changes.

Apply CP is appropriate in a staging environment, where you know all changes have been tested and can be propagated to the next stage as a group. Resync CP is appropriate for situations where you want to select individual changes, and build and test the changes in your Sandbox before propagating them.

The effectiveness of Apply CP and Resync CP relies on a change package methodology that includes the following practices:

- accurate and consistent use of change packages for logging items
- associating related changes into a single change package that addresses the item in question

Practices to avoid include the following:

- including changes made on multiple variants in the same change package
- including unrelated changes in a change package

Apply CP and Resync CP are most useful for code and other text files where differencing can be performed. The operations are not recommended for binary files because of the difficulties encountered in differencing and merging binaries.

---

**NOTE** For brevity, some of the Apply CP and Resync CP examples use the command line interface to illustrate how a command works. For information on using Apply CP (`si applycp`) and Resync CP (`si resynccp`) in the command line interface, see the *MKS Integrity 2009 CLI Reference Guide for Configuration Management*.

---

# Advanced Apply CP and Resync CP Concepts

This topic contains advanced concepts that may be of use when using Apply CP and Resync CP in more complex scenarios.

## Understanding Net Changes

The intent of the Apply CP and Resync CP commands is to propagate the net changes encapsulated in a set of change packages from the source project tree to a specified target project tree.

*Net changes* are a set of changes that include both the content of members and the structure of the tree itself (the tree structure is defined by the subprojects in the tree).

For example, John Riley, a developer, is propagating a set of change packages that includes two checkin operations for member `ultim.txt` followed by a drop operation for `ultim.txt` in the source tree. MKS Integrity determines the net change is a drop of `ultim.txt` and so only performs that drop operation on the target; ignoring the preceding checkins.

## Operations Propagated

Only changes to member content and subproject structure are propagated by the command to the target project tree. All other types of changes (such as configuring subprojects, adding member attributes, freezing members, and so forth) are not propagated by the command.

In general, only operations explicitly recorded in the change packages being propagated (which includes those change packages picked up by the backfill) are propagated from the source project tree to the target project tree. However, there are situations where certain operations may not have been recorded in a change package, and so consequently MKS Integrity identifies those as *implicit operations* that must propagate to the target tree.

Implicit operations are largely limited to subproject operations (because prior versions of MKS Integrity did not include subproject operations in change packages). For information on controlling if such implicit operations are allowed, see the **Propagate Subprojects** option for the Apply CP command in “Apply CP Procedure” on page 433 or for the Resync CP command in “Resync CP Procedure” on page 446. For additional information relating to implicit operations, see “Using the Apply CP Backfill List” on page 429 and “Using the Resync CP Backfill List” on page 439.

## How Net Changes Are Calculated

Net changes are determined by calculating a single operation for each member or subproject that represents the net change for that member or subproject in order to make the target project equivalent to the source project.

For the Apply CP command, the net change needs to correspond to an existing artifact in the repository. For example, for a member, the net change needs to correspond to an existing revision in that member’s archive (unless the net change is to drop that artifact). If it is impossible to resolve the net change into an existing artifact (possibly because a merge is required) the propagation fails and you (the user) must use the Resync CP command to perform the task instead.

The Resync CP command also calculates a single operation for each member or subproject, but unlike Apply CP, Resync CP operates in the context of a sandbox and allows merges to take place in the

---

sandbox which will be checked in to create a new revision in the member's archive when the propagation change package is submitted. However, there are limits on how complex the merge can be. For example, if generating the single operation for a member requires merging in (or merging out) two or more discontiguous sets of revisions from the member's archive, the operation fails. It is still possible to propagate the changes in most cases, but the propagation needs to be broken up into smaller parts and performed sequentially.

### ***Implications When Net Change Exists in Target***

An important factor in understanding how MKS Integrity determines net changes is that MKS Integrity considers a change as already having been propagated from the source to the target if the member revision in the target is already at (or beyond) the revision that the net change indicates should be propagated. This specifically has implications in two cases: first, when you try to re-propagate a change from the source project tree to the target project tree; and second, when you try to propagate a change that moves a member revision backward along its history. In both cases, MKS Integrity determines that the target already has the required change, so it performs no action. The following is an example:

John, a developer, has member `bar.txt` at revision 1.6 in the source project hierarchy, but that same member is at revision 1.5 in the target project hierarchy. In the source hierarchy, John creates a new change package 1:1 containing an entry which does an Update Member Revision operation of `bar.txt` back to revision 1.4. If another user, Mary, tries to propagate this change package to the target using Apply CP (or Resync CP), MKS Integrity does nothing since the target already has the change (since by implication revision 1.5 is built on top of 1.4). If Mary wants to force the member revision back to 1.4 in the target, she must do so manually and cannot rely on Apply CP (or Resync CP) to do it for her.

This behavior of determining which changes the target "already has" is based on revision numbering. Ignoring those changes is fundamental to the command process and has impact on the backfill process as well.

### ***Changes Used When Propagating***

Another important factor in understanding how MKS Integrity determines net changes that needs to be understood is that only changes that actually affect the project's member and subproject configuration are taken into account when propagating. For example, it is possible to check in new revisions to an archive using a change package and at that time indicate that these new revisions should not update the member revision in the project. When propagating this change package to the target, those entries are ignored by the command since they do not update the member configuration in the source project.

### ***Net Changes Grouped in Virtual Bucket***

MKS Integrity handles the net changes by internally creating a virtual *bucket* for each member or subproject that it encounters in the list of change packages to be propagated (including the changes explicitly specified by you (the user) and the ones you specify to pick up in the backfill). MKS Integrity examines all of the change package entries in those change packages one at a time in the order in which they originally occurred, and then adds the entry to the appropriate bucket. The changes are grouped in the following way:

- For members, the primary identifier of the bucket is the location of the member's archive in the repository.
- For subprojects, primary identifier of the bucket is the canonical location of the subproject's underlying project in the repository.
- For moved (and renamed) members and subprojects, the bucket tracks both the old and new location (and name) of the member or subproject being moved (or renamed).

---

For MKS Integrity to be able to propagate the changes from the source project tree to the target project tree, MKS Integrity must locate (or if necessary add or create) the corresponding target member or subproject for each source member or subproject.

For subprojects, if MKS Integrity determines that a subproject is missing in the target, the corresponding variant subproject will be created in the target immediately during the initial phase of calculating the net changes to be propagated (although you the user are prompted in advance that it needs to do so and asked if it is ok to proceed). Variant subprojects created in this way remain in the target hierarchy even if you later decide to cancel the propagation; a limitation of MKS Integrity that you need to understand. All other changes to the target project tree are only committed if and when you have had a chance to examine the set of net changes to be preformed and you agree to proceed with the propagation.

## Problems Associated With Propagating Changes

Many of the problems associated with propagating changes involve moved and renamed members, and moved subprojects. The problems fall into two main categories:

- Locating the corresponding member or subproject in the target when the member or subproject in the source has been moved or renamed
- Processing entries in the best order when multiple move operations have taken place (especially important with nested move subproject operations)

### ***Locating Corresponding Member or Subproject in Target Source Member or Project Was Moved or Renamed***

The following is a general description of the process MKS Integrity uses to locate the target project for a given change package entry:

- 1 Given the path to the source project that owns the change package entry, MKS Integrity determines if the target project (which is a different variant of this project) already exists in the target hierarchy. If the target project exists, use it as the target.
- 2 If the target project does not exist, see if the source project is still visible (registered); in other words, the source project has not been moved or dropped. If the source project does exist, create the corresponding variant subproject in the target hierarchy.
- 3 If the source project does not exist, it may be because it was moved. Use the configuration path encoded in the change package entry to try to find the source project's location after the move<sup>1</sup>. If MKS Integrity finds the source project, find or create the corresponding project in the target project tree.
- 4 If the previous attempts fail, no further attempts are made. MKS Integrity cannot find the target project to propagate the changes to. You must manually perform the propagation.

---

**NOTE** This section is a major simplification of the logic used by MKS Integrity, and is only intended to provide a general idea of the process. There are many details omitted that are beyond the scope of this documentation.

- 
- 5 Once the target project has been located, MKS Integrity attempts to find the target member or subproject within that target project that the changes need to be propagated to.

For all operations other than move and rename, MKS Integrity looks for the member (or subproject) in the target project tree with the same name as the member (or subproject) in the

---

<sup>1</sup> The configuration path information is only available for changes package entries created by MKS Integrity Server 2009 or later. For change package entries created prior to that release, this step is not performed.

---

source project tree (verifying that the member or project has the same backing archive or project). If MKS Integrity finds the corresponding target, it updates it accordingly. Otherwise MKS Integrity adds the member (or subproject) to the target project tree (pointing to the same backing archive or project).

For moves and rename operations, MKS Integrity Server follows the same basic process as other operations, but with the addition that MKS Integrity Server knows both the "from" and "to" names for the move or rename operation, so it is more intelligent in searching for the target.

### ***Processing Entries in Best Order When Multiple Move Operations Exist***

As previously stated, MKS Integrity generally examines entries within a bucket in chronological order, with earlier entries processed before later ones.

---

**IMPORTANT** MKS Integrity is not propagating changes sequentially. It is examining them sequentially to produce the single net result for that bucket.

---

The net results are grouped by the parent project that they belong to and the corresponding operations are performed on each parent project as a single bulk operation on the MKS Integrity Server, as whenever possible.

In cases where operations are being propagated across multiple projects (or subprojects), the MKS Integrity Server uses an optimal ordering in which to process each project's corresponding bulk operations. As of MKS Integrity 2009, the configuration path information is used to order the processing of the projects (or subprojects) so that if they exist in a hierarchy parent projects are processed before their child subprojects. This has been empirically shown to provide better results in situations where complex subproject refactoring scenarios (such as multiple nested subproject moves) are being propagated.

## **Making Propagations More Successful**

The following are general rules that you can follow to make propagations easier, less error prone, and more likely to succeed:

- Perform many small propagations rather than a few large ones.
- Wherever possible, propagate changes to the target project in the same order that they were originally created in the source project.
- When refactoring operations are involved (such as move and rename operations on members or subprojects) the previous rules are especially important. If possible, process refactoring operations in small chunks in the same order in which they originally occurred.
- Because subproject operations are not detected by the backfill algorithm, always try to explicitly specify change packages containing subproject operations, rather than depending on the backfill algorithm to locate them for you.
- Whenever possible, always propagate changes in the same direction. For example, one project (or variant) is always the source project tree and the other is always the target project tree when propagating changes between the two trees. Maintaining the same propagation direction helps the backfill algorithm keep track of what has (and has not) already been propagated between the two development paths.
- Whenever possible, keep changes to different variants in different change packages. In other words, the changes in a single change package should all correspond to the same development path (or the mainline project).

# Resync By CP Overview

The Resync By CP command is primarily a tool for developers. When you want to resynchronize files in your Sandbox, you usually do so by choosing individual files and then using the Resynchronize (`si resync`) command. However, if the files you are resynchronizing have changes that are linked to other files, the standard resync operation would not include those related files. To resynchronize all related files, you would have to manually search for all the changes associated with the change package on the member you are resynchronizing.

The Resync By CP command automates this process by searching the change package specified on the member revision you are resynchronizing and then bringing the changes from the project to your Sandbox.

While the Resync CP command searches all files related to a selected change package, and all the change packages that may be associated with the related files, the Resync By CP command only processes the change packages associated with the member you are resynchronizing.

To Resync By CP, from a **Sandbox** view, select one or more members that contain member deltas, and select **Member > Resynchronize By Change Package**.

Depending on the preferences you have set for the Resynchronize command, when you Resync By CP, the **Confirm Overwrite Working File** dialog box displays. If you want to retain your changes in the working file, click **No** (**No to All** for multiple members). If you want to compare your working file with the revision you are resynchronizing it with, click **Differences**. To merge and resynchronize the member, click **Yes** (for multiple members, click **Yes to All**).

## **How Does the Resync By CP Command Work?**

In a Resync By CP operation, the change package list is computed based on the member you are resynchronizing (whereas in a Resync CP operation, you explicitly state the list of change packages).

The functioning of Resync By CP is affected by the settings you choose for the Resync CP command under **File > Edit Preferences**. This includes the way the backfill list operates. For example, if you specify ask, a backfill list displays.

## **When Should I Use the Resync By CP Command?**

Developers should use Resync By CP when they want to ensure that they have all dependent changes associated with a member revision, even if these changes are contained in other files. For example, a developer needs to check out (locked) a file and modify it. The developer finds that other revisions have been checked in since the member was resynchronized in the Sandbox. Because the Sandbox is quite large and contains many unrelated changes, the developer does not want to perform a standard resynchronization. The Resync By CP option can be used in this situation.

---

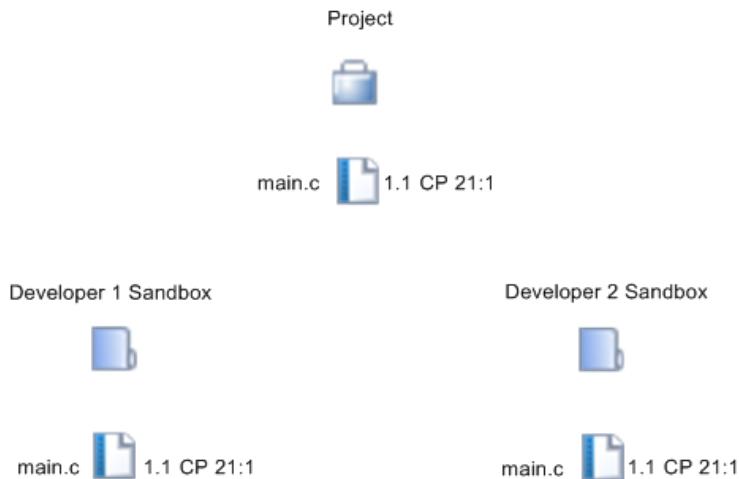
**NOTE** The functioning of Resync By CP is affected by the settings you choose for the Resync CP command under **File > Edit Preferences**. The Resync By CP operation always sets the backfill list to **Entire Change Packages** (`cp`).

---

## **Example of Resync By CP**

Consider a case where a developer makes a change to project member `main.c`, and that change requires an additional file, `main.h`. A standard resync operation for `main.c` would not capture `main.h`.

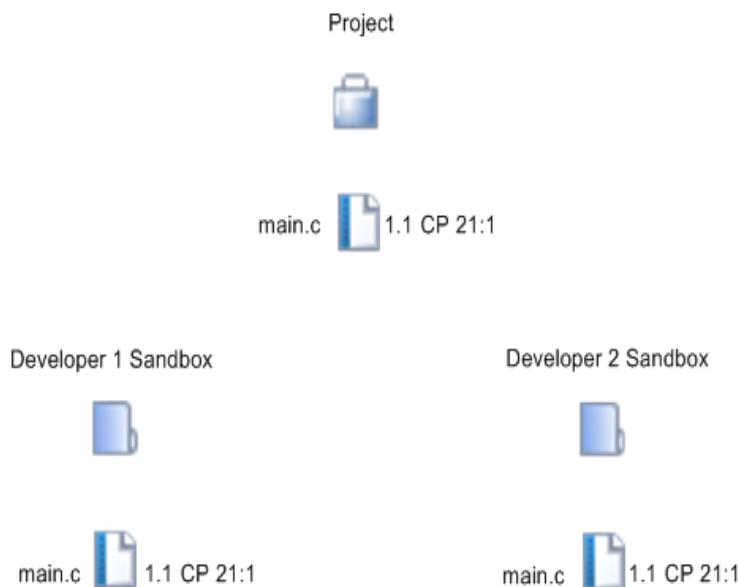
In the initial stage, Sandboxes pointing to the project include `main.c` at revision 1.1.



*Before using Resync By CP in a development environment*

Developer 1 then performs the following tasks:

- checks out and locks `main.c`, revision 1.1
- makes an update to `main.c` that requires the `main.h` file
- checks in the changes to `main.c` and associates these changes with CP 22:1
- also against CP 22:1, adds `main.h` as a member of the project



*After using Resync By CP in your Sandbox to capture all changes (including new files) contained in the associated change package*

When Developer 2 uses the Resync By CP command to resync `main.c`, his Sandbox is updated to show that `main.c` is at 1.2 and that `main.h` has also been added to the project as part of CP22:1.

---

**IMPORTANT** If the working file of the member you are resyncing is modified, MKS Integrity asks you to confirm that you want to merge your modifications into the working file.

---

---

# Integrating With Implementer

Implementer provides host-based version control and configuration for software management. It also provides promotion and deployment for the movement of software changes to QA and production. For user information on Implementer, see the *MKS Implementer User Guide*. For administrative information on Implementer, see the *MKS Implementer Installation and Administration Guide*.

An Implementer change package reflects the iSeries development done to resolve the item. Integrating MKS Integrity with Implementer allows you to:

- track each member, how it was changed, and where it is currently in the development process
- group changes by the associated production environment
- view Implementer information from the MKS Integrity Web interface

The following outlines some of the main considerations when using MKS Integrity items and Implementer change packages. Each change package reflects the members created or changed for a single production environment.

- Multiple change packages are created for a single item when multiple production environments are affected, for example, when making a software fix to an older version as well as to the new release. If an item needs to be fixed in more than one application, a new change package is created for each application keeping each fix separate.
- Development progress in Implementer optionally updates the state of an item to reflect the development progress within Implementer. For example, when a member is checked out for an item, the item state could be automatically set to `Coding`. When all changes for an item are promoted to quality assurance, the state could be automatically set to `Testing`.
- Implementer development activities are controlled by an item's workflow.

## ***Example of Implementer Integration***

The following example explains how the integration between Implementer and MKS Integrity works:

- 1 An item is created in MKS Integrity. For example, the item requests a bug fix for a software installation problem.
- 2 An Implementer change package is created automatically by performing a check out within Implementer. Likewise, a change package is automatically created from the Workbench by assigning an item to a lock in the **Lock Detail** panel or by assigning an additional item in the **Multiple Items** panel. The change package is assigned an ID and is in an *open* state.
- 3 Once the changes or additions necessary to satisfy the item are complete, the user promotes the members.
- 4 The item is moved to the state in the workflow defined for the environment. For example, the item is moved from state `Test` state to state `QA1`.

---

# Index

## A

add

- attachment to an item 31
- expression
  - to distribution chart 91
  - to item fields chart 91
  - to item fields trend chart 95
- query filter to a query
  - in the Web interface 69
- related item to an item 29
- rich content to an item 24
- time entry to an item 57

aggregate computation

- in distribution charts
  - creating 91
  - described 86
- report, creating 122

aggregated function

- described 165

aggregation expression

- described 159

annotated revision 374

apply change package 432

- command options 436
- concepts 427
- example 426
- overview 451
- procedure 433
- using 426
  - using a propagation CP 431
  - using the backfill list 429
  - with no dependencies 428

apply rich content 24

archive

- described 360
- filter by member with shared archive 315
- information view 366

arithmetic calculation

- creating for item fields chart 91

arithmetic function

- described 165

arithmetic operators

- computed expression 162

attachment

- filtering queries by 66
- key considerations 31
- operations 32
- working with 31

attribute

- outline 187
- project 243

Author keyword 351

axes, specifying for chart 97

## B

backfill list

- using with Apply CP 429
- using with Resync CP 439

base time report

- described 134

running 134

baselining

- documents 210
- multiple documents 210

batch edit 36

- item 36

blocks

merging

- individual 389
- nonconflicting 389
- using Visual Difference 386

boolean operators

- computed expression 162

branch

filtering queries by 66

member, overview of 381

merge

- by dragging 382
- on check out 383
- on resync 383

renaming member on branch 339

starting during check in 336

viewing details 214

viewing historical items based on 42

branch, filter for members on 315

branching

- documents 212

build sandbox

- creating 291

burn-down chart 95

## C

change package

adding entries 400

advantages of using 397

apply

- command options 436
- concepts 427
- example 426
- overview 451
- procedure 433
- resolving conflicts 432
- using 426
  - using a propagation CP 431
  - using the backfill list 429
  - with no dependencies 428

closing 415

creating 400

described 396

discarding 416

editing 409

entries

differences 403

discarding 402

moving 410

filtering queries by 66

finding 404

generic change package view 412

generic, viewing 412

Implementer 458

integrating with workflows and documents using 396

---

overview 396  
pending operations in 421  
propagation, using with resync cp 443  
resynchronize  
    command options 448  
    example 438  
    overview 451  
    procedure 446  
    using 438  
    using the backfill list 439  
resynchronize by  
    example 456  
    how it works 456  
    using 456  
review 422  
    accepting change packages 422  
    benefits of 418  
    entry categories 420  
    overview 417  
    rejecting change packages 423  
    reopening change packages 425  
    review log 408  
    steps 417  
    workflow 418  
specifying for an operation 400  
submitting 414  
using to control development 397  
view 407  
    viewing details and entries 406  
    viewing your change packages 399  
character encoding 310  
chart  
    burn-down 95  
    creating 89  
        distribution chart 90  
        item fields chart 91  
        item fields trend chart 94  
        trend chart 92  
    deleting 114  
    described 86  
    detailed examples of 99  
    editing 109  
    manage charts view 87  
    printing 113  
    properties 109  
    running  
        in the GUI 110  
        in the Web interface 111  
    saving 113  
    specifying  
        axes 97  
        graph 96  
        legend 97  
        ranges 98  
        table graph style 97  
check in member 334  
check out  
    automatic merging on check out 383  
    member 329  
checkpoint project 267  
colours  
    defining in document view 186  
command, ident 351  
CompanyInfo keyword 351  
compare  
    member differences 377  
    two text files 379  
    working file to member revision 377  
comparison report  
    running 134  
computed expression  
    described 159  
    function classes 165  
    group fields 164  
    rules 161  
    types 159  
    user fields 164  
conflicts  
    resolving when editing an item 37  
content  
    about 181  
    adding to documents 194  
    copying 199  
        for reuse 200  
        historical 200  
    inserting item 194  
    modifying 199  
    restricturing 201  
copy  
    dashboard 153  
    query 79  
create  
    chart 89  
    dashboard 143  
    item 21  
    query 68  
    time entry 57  
cycle detection for related items 3

## D

dashboard  
    applying a project filter to 151  
    clicking a distribution chart 152  
    clicking a query link 152  
    clicking a report link 152  
    clicking a URL link 152  
    components 138  
    copying 153  
    creating 143  
    dashboard definition view 140  
    deleting 154  
    described 137  
    designing 138  
    editing 150  
    layout designer interface 141  
    running 151  
Date keyword 351  
date, viewing historical items based on 42  
day/time function  
    described 165  
deferred member, described 391  
deferred operation 391  
    adding member from archive 309  
    canceling 359  
    filter by members with 315  
    import 317  
    resynchronizing 358  
    submitting 357  
    updating a member's revision 369, 372  
delete  
    chart 114  
    dashboard 154  
    labels 40  
    query 80  
    report 133  
    time entry 59

---

**demote**  
 member 393  
 project 283  
**designing a dashboard** 138  
**development path**  
 creating 274  
 described 274  
 removing 275  
**difference**  
 change package entries 403  
 compare checkpoints 269  
 compare members 377  
 compare to last checkpoint 269  
 compare to specific checkpoint 269  
 project differences view 269  
 two text files 379  
**Visual Difference**  
 editing merge results 386  
 merging 385  
 reverting merge results 387  
 saving merge results 387  
 working with 385  
 working file and member revision 377  
**disable**  
 keywords 350  
**distribution chart**  
 aggregate operations  
     described 86  
 creating 90  
     described 86  
**document**  
 adding content 194  
 baselining 210  
     individual 210  
     multiple 210  
 branch 212  
     propagating traces to 218  
     viewing branch details 214  
 changing project 209  
 configuring Document view 185  
 content  
     copying 199  
     modifying 199  
 copy,cut,paste,move,drag 186  
 creating 192  
     from template 193  
 export 224  
 filtering queries by content items used in the  
     document model 66  
 finding 189, 203  
 historical 203  
 import 193, 224  
 included and referenced 194  
 inline editing 185  
 inserting segment 195  
 messages 186  
 moving content around 201  
 opening 203  
 relationships, viewing 208  
 reports  
     current 217  
     historical 217  
 restructuring content in 201  
 section, deleting 201  
 segments  
     including 196  
     inserting nodes into 196  
 structure  
 terms and concepts 181  
 tooltip display 186  
 view as of  
     branch 203  
     label 203  
     specific date 203  
     user edit 203  
 view by query 203  
 viewing and editing properties 207  
**document item** 183, 184, 192  
**Document view**  
 configure 185  
 pinning a filter 18  
**document view** 184  
 about and graphic 184  
 indicators and icons 190  
 outline 185  
**documents**  
 function classes 166  
**drag-and-drop, relationships** 29, 46  
**drop project or sandbox**  
     described 247

**E**

**edit**  
 chart 109  
 dashboard 150  
 item 34  
     conflicts 37  
**items**  
     Items view 33  
     Relationships view 33  
**query**  
     properties 77  
**report** 129  
     time entry 59  
**edit history**  
     viewing historical items based on 42  
**electronic signature** 32  
**e-mail hyperlink** 10  
**e-mail notification**  
     described 50  
     setting 50  
         in the GUI 51  
         in the Web 52  
**empty fields**  
     computed expression 163  
**encoding, character** 310  
**encrypted archives** 245  
**expansion**  
     keyword  
         turn off 350  
**export**  
     document 224  
     items to Microsoft Excel 81  
**expression, creating for**  
     distribution chart 91  
     item fields chart 91  
     item fields trend chart 95  
**external information function**  
     described 165

**F**

**field**  
     batch editing 36  
     filtering queries by 66  
**fields** 22  
**filter**

---

adding to a query  
    in the Web interface 69  
all members 314  
any member deltas 314  
applying a project filter to a dashboard 151  
attachment filter type 66  
branch filter type 66  
change package filters 66  
deferred operations 315  
document model items 66  
existing working files 314  
field filter type 66  
frozen members 314  
history filter type 66  
label filter type 66  
locked members 314  
member archive shared 315  
member history 364  
members at label 314  
members at state 314  
members locked by me 314  
members on branch 315  
members with attribute 314  
members with label 314  
members with name 315  
members with rule 315  
missing working files 314  
modified working files 314  
nesting in queries 67  
new members 314  
new revision available 314  
out of scope members 314  
out of sync members 314  
pending operations 315  
pinning 18  
project filters 17  
relationship filter type 66  
sparse sandbox contents 314  
sub-query filter type 67  
test results filter type 67  
time entry filters 67  
unresolved merges 314  
work in progress 314  
working file on branch 314  
working file size deltas 314

finder dialog box 189  
    options 203

function

- aggregate  
    described 165
- arithmetic  
    described 165
- day/time  
    described 165
- external information  
    described 165
- item information  
    described 165
- test  
    described 166
- text  
    described 165

function classes  
    computed expression 165

## G

generic change packages, viewing 412  
graph, specifying for a chart 96

group fields  
    computed expression 164  
group, filters in queries 67

**H**

Header keyword 351  
historical content 203  
    copying 200  
historical item  
    viewing as of 42  
    viewing current information 43  
historical report  
    described 134  
    running 134  
history, filtering queries by 66  
HTML  
    applying to items 24  
hyperlink  
    e-mail 10

**I**

Id keyword 351  
ident command 351  
Implementer, integrating with 458  
import  
    documents 193, 224  
    member 316  
    project 245  
    sandbox 297  
    shared 299  
included and referenced documents  
    differences 194  
inline edit 33  
integration  
    with Implementer 458  
intra-item expression  
    described 159  
item  
    adding  
        attachment 31  
        related item 29  
        rich content 24  
        time entry 57  
    batch editing 36  
    compare to base time report 134  
    content  
        editing 197  
        viewing 197  
    copying 41  
    create  
        inline 33  
        creating 21  
        differences report 135  
    document  
        editing 197  
        viewing 197  
    edit  
        inline 33  
        editing 34  
        conflicts 37  
    electronic signatures for 32  
    exporting to Microsoft Excel 81  
    fields 22  
    filtering queries by 66  
    historical  
        key considerations for viewing 42  
        viewing 42  
    information on tabs 9

---

<p>printing 49 relationships     cycle detection for 3     field behavior 3     overview of 3     Relationships view 44 removing related items 30 report     current 217     historical reports 217     running 134 tabs     Attachments 11     Branches 11     Change Packages 11     Fields 11     History 12     Labels 11     Relationships 11     Time Entries 11     Workflow 12 working with test results for 231 item differences report, running 135 item fields chart     creating 91     described 86 item fields trend chart     burn-down chart 95     creating 94     described 86 item information function     described 165 Items view 5     configuring 7     drag-and-drop, relationships 29     editing items 33     pinning a filter'Relationships view         pinning a filter 18     Web modes 7</p>	<p>to member 353 to project 265 deleting 40     from member 354     from project 266 filter member with label 314 filtering queries by 66 moving 40     viewing historical items based on 42 legend, specifying for a chart 97 line terminator 290 locate view 281 lock 295     described 344     downgrading 347     member 344     removing 347     view locks 345     viewing 345 locked member filter 314 Locker keyword 351 Log keyword 352 log, time entry 57</p>
<p><b>K</b></p> <p>keyword     Author 351     CompanyInfo 351     Date 351     Header 351     Id 351     locating 351     Locker 351     Log 352     Name 352     ProjectLabel 352     ProjectName 352     ProjectRevision 352     ProjectSetting 352     RCStfile 352     Revision 352     SandboxSetting 352     Setting 352     Source 352     State 352     table of 351     turning off 350     using 349</p>	<p><b>M</b></p> <p>member     adding from archive 308     adding label to 353     adding to project 305     attribute         editing 318         viewing 318 branch     overview of 381 checking in 334 checking out 329 deleting label 354 demoting 393 dropping 311 editing 328 filter frozen members 314 filtering 313 former, described 311 freezing 355 history, filtering 364 importing 316 information     editing 318     viewing 318 label     editing 319     viewing 319 locating where used 280 locking 344 making working file writable 348 merging     overview of 381 moving from project 322     via dragging 324 operations     deferring 391     pending 421     submitting deferred 357 promoting 393 removing from project 311 renaming 339 resynchronizing 343 reverting 341, 342</p>

## L

label  
    adding 40

---

revision, updating 367  
 rule  
     editing 319  
     viewing 319  
 rule, setting 370  
 selecting 320  
 thawing 356  
 unlocking 347  
 updating 343  
 view sandbox that locked member 295  
 viewing content of 328  
 viewing locked 345

**merge**  
 automatic merging on check out 383  
 automatic merging on resync 383  
 blocks using Visual Difference 386  
 by dragging 382  
 in Visual Difference 385  
 member  
     overview of 381  
 results  
     editing in Visual Merge 389  
     reverting in Visual Merge 390  
     saving in Visual Merge 390  
 revision  
     in Visual Difference 385  
     in Visual Merge 388  
 suspending in Visual Merge 390

metrics, project 271  
 Microsoft Excel  
     export items 81  
 MIME type 66  
 MKS Visual Difference  
     See Visual Difference  
 moving 40

**N**

Name keyword 352  
 named query, described 62  
 nesting filters in queries 67  
 nodes  
     about 181  
 non-member, displaying 326  
 notification  
     setting e-mail notifications 50  
         in the GUI 51  
         in the Web 52  
 notifications  
     e-mail  
         setting up on web UI 52  
 numeric field  
     creating for item fields trend chart 95

**O**

operations  
     document view 190  
 operations mode 186  
 outline attributes, defining 187

**P**

pending  
     operations 421  
     revision  
         reviewing 376  
         working with 376  
 pending, filter pending member operations 315  
 pin

filter 18  
 portfolios functionality 44, 134  
 print  
     chart 113  
     labels 40  
     report 132  
 project  
     adding 248  
     adding member 305  
     adding members from archive 308  
     attribute 243  
     branching 273  
     checkpointing 267  
     configuration for sandbox 301  
     creating 244  
     demoting 283  
     difference  
         compare checkpoints 269  
         compare to last checkpoint 269  
         compare to specific checkpoint 269  
         view 269  
     dropping 247  
     history  
         view 261  
     importing 245  
     importing members into 316  
     information view 242  
     label  
         adding 265  
         deleting 266  
     locating where used 280  
     metric 271  
     opening 249  
     projects view 241  
     promoting 283  
     removing members 311  
     restoring 263  
     view  
         GUI 250  
         viewing differences 269

project filter  
     described 17  
     using 17  
 ProjectLabel keyword 352  
 ProjectName keyword 352  
 ProjectRevision keyword 352  
 projects  
     document  
         changing 209  
 ProjectSetting keyword 352  
 promote  
     member 393  
     project 283  
 propagating traces  
     example 218  
 propagation change package  
     using with apply cp 431  
     using with resync cp 443  
 properties  
     charts 109  
     query 77  
     report 129  
     viewing and editing document 207

**Q**

queries  
     using to find documents 203  
 query

---

<p>adding a filter to     in the Web interface 69</p> <p>attachments filter type 66</p> <p>branch filter type 66</p> <p>builder     in the GUI 65     in the Web interface 65</p> <p>change package filter type 66</p> <p>copying 79</p> <p>creating 68</p> <p>deleting 80</p> <p>described 62</p> <p>editing 77</p> <p>entering general information for     in the GUI 68</p> <p>field filter type 66</p> <p>history filter type 66</p> <p>item type 66</p> <p>label filter type 66</p> <p>manage queries view 63</p> <p>nesting filters in 67</p> <p>pinning a filter 18</p> <p>query definition view 64</p> <p>relationships filter type 66</p> <p>running 78</p> <p>sub-query filter type 67</p> <p>test results filter type 67</p> <p>time entry filter type 67</p> <p>using project filters     in the GUI 17</p> <p>query results     export to Microsoft Excel 81</p> <p>quick query, described 62</p>	<p>rename member 339</p> <p>report     compare to base time reports described 134     deleting 133     described 116     editing 129     historical reports described 134     manipulating data in Excel 132     printing 132     properties 129     running         compare to base time 134         comparison 134         current 217         historical 134, 217         item differences 135         item reports 134         saving 132</p> <p>resolving conflicts 432</p> <p>resynchronize     automatic merging on resync 383</p> <p>by change package     example 456     how it works 456     using 456</p> <p>change package     applying change packages from two variants 442     applying changes from a variant 441     binary conflicts 445     command options 448     example 438     overview 451     procedure 446     using 438     using the backfill list 439</p> <p>member 343</p> <p>members with deferred operations 358</p> <p>reuse reference mode     changing 201</p> <p>reusing content     copying 200</p> <p>revert     member 341, 342     merge results in Visual Difference 387     merge results in Visual Merge 390</p> <p>review     change package         benefits of 418         entry categories 420         overview of 417         steps 417         workflow 418     pending revision 376</p> <p>revision     annotated 374     assigning descriptions 334     character encoding 310     merging         in Visual Difference 385         in Visual Merge 388     number, assigning 335     pending         reviewing 376         working with 376     updating member revision 335</p> <p>Revision keyword 352</p> <p>rich content     HTML elements and attributes 27</p>
--	--

---

<p>key considerations 25 working with 24</p> <p>root documents   create 192</p> <p>row   inserting 194</p> <p>rule   filter by member with rule 315</p> <p>rules   computed expressions 161</p> <p>run   dashboard 151   item report 134   query 78</p>	<p>shortcut keys   document view 190</p> <p>snapshot sandbox 302</p> <p>Source keyword 352</p> <p>sparse sandbox contents, filter 314</p> <p>State keyword 352</p> <p>submit   change packages 414   deferred member operations 357</p> <p>subproject   adding 253   components 252   configuring 256   creating 252   dropping 258   locating where used 280   moving 259   pending operations 421   shared, adding 254</p> <p>sub-query   described 62   filtering queries by 67</p> <p>subsegments   toggle include/insert 201</p> <p>suspect traces 220</p> <p>syntax   computed expression 162</p>
<p><b>S</b></p> <p>Sandbox   scope definition 291</p> <p>sandbox   creating 291   described 286   dropping 300   importing 297   opening 295   retargeting 301   scanning for change 328   shared 298     configuring 299     importing 299     removing 299</p> <p>snapshot 302</p> <p>types of 286</p> <p>variant 286</p> <p>view information in graphical user interface 295</p> <p>viewing contents 295</p> <p>viewing information for 289</p> <p>viewing sandbox that locked member 295</p> <p>viewing sandboxes 288</p> <p>SandboxSetting keyword 352</p> <p>save   chart 113   report 132</p> <p>scope definition   Sandbox 291</p> <p>searching text   DB2 14   in the GUI     using a project filter 16</p> <p>MS SQL 15</p> <p>operators 13, 14, 21</p> <p>Oracle 15</p> <p>segment   about 181   including a new or existing 196   inserting 195     a node into open segments 196</p> <p>root   making visible 201</p> <p>Setting keyword 352</p> <p>shared items   about 181</p> <p>shared reference mode   changing 201</p> <p>shared sandbox 298   configuring 299   importing 299   removing 299</p> <p>shared subproject, adding 254</p>	<p><b>T</b></p> <p>table graph style, specifying for a chart 97</p> <p>table of keywords 351</p> <p>template   using document as 193</p> <p>test function   described 166</p> <p>test result editor 234   configuring 236   editing test results 237   entering test results 229   interface 234</p> <p>test results   editing 237   entering 229   filtering queries by 67</p> <p>tabs   Attachments 228   Relationships 228   Verdict 228</p> <p>test result editor 234</p> <p>viewing 233</p> <p>viewing details for 228   tabs 228</p> <p>working with for an item   test results tab 231</p> <p>text function   described 165</p> <p>text search   DB2 14   filtering by project 13   in the GUI 16</p> <p>MS SQL 15</p> <p>operators 13, 14, 21</p> <p>Oracle 15   saving 14</p> <p>thaw member 356</p> <p>time entry   creating 57   deleting 59</p>

---

- editing 59
- filtering queries by 67
- viewing 60
- tip revision, renaming 339
- trace
  - creating 215
- trace relationships
  - create 215
  - drag-and-drop creation 215
  - manage 215
- traces
  - example
    - propagate to branch 219
    - propagating 218
  - propagate
    - procedure 220
  - propagating
    - how 219
  - propagating to branched documents 218
  - propegate procedure 220
  - suspect 220
  - viewing for content items 216
- trend chart
  - creating 92
  - described 86
- turn off
  - keyword expansion 350

## U

- unlock, member 347
- user fields
  - computed expression 164
- UTF-8 310

## V

- variant project
  - applying changes from 441
  - applying changes from two variants 442
- variant sandbox
  - creating 291
  - described 286
  - using 286
- view
  - branch details 11
  - dashboard definition view 140

- generic change package 412
- generic change package view 412
- historical item 42
- item details in the GUI 9
  - tabs 9
- Items view 5
- items view in the Web interface
  - query builder 65
- Items view modes in the Web interface 7
- manage charts view 87
- manage queries view 63
- query definition view 64
- Relationships view 44
- test result details 228
- test results 233
- time entry 60
- viewing or editing panel
  - described 185
- Visual Difference
  - editing merge results 386
  - merging 385
  - reverting merge results 387
  - saving merge results 387
  - working with 385
- Visual Merge
  - editing merge results 389
  - merging revision 389
  - reverting merge results 390
  - saving merge results 390
  - suspending merges 390
  - viewing conflicts 388
  - working with 388

## W

- Web interface
  - registered projects view 241
- web interface
  - notifications
    - setting up e-mail 52
- workflow, change package reviews 418
- working file
  - comparing to member revision 377
  - discarding change 341, 342
  - making writable 348