

Jira Interface for MDCMS

from Midrange Dynamics

From Version 8.6

Published December 5, 2023

Overview

What is Jira?

Jira is a web-based, proprietary issue management product, developed by Atlassian. It can be licensed to run in the cloud on Atlassian servers or licensed as a server product installed within the organization's infrastructure. The Jira interface for MDCMS is designed to communicate seamlessly with either licensing version with a common set of administrative tools.

MDCMS interfaces with Jira Software, which provides project management for development teams and with Jira Service Desk, which provides issue tracking. Both products use the concept of projects and issues within those projects and both products share the same core data elements which can then be shared with MDCMS.

Information Shared between Jira and MDCMS

The intention of the interface is to share data and workflow control between MDCMS and Jira in order to avoid needing to manually duplicate entries in both systems as well as to be able to manage issues from either system.

Some Project Types or Task Types can be defined as being applicable to MDCMS or Jira as well as specifying where issues of the given type may originate. This allows for the flow of information to differ by type and to reduce the amount of information shared between systems to only that which is relevant.

Table of Elements passed from Jira to MDCMS

Jira Element	MDCMS Element
Project Key	Project ID
Project Category	Project Type
Project Description	Project Title/Description

Jira Element	MDCMS Element
Project Owner	Project Requester
Issue Key	Task Reference Code
Issue Type	Task Type
Issue Status Code	Task Status Code
Issue Priority	Task Priority
Issue Due Date	Task Due Date
Issue Parent Task	Task Parent Task
Issue Assigned to User	Task Assigned to User
Issue Creator	Task Requester
Issue Create Date	Task Create Date
Issue Create Time	Task Create Time
Issue Hours Expected	Task Hours Expected
Issue Summary	Task Summary
Issue Description	Task Description

Table of Elements passed from MDCMS to Jira for Issues originating in Jira

MDCMS Element	Jira Element
Task Status Code	Issue Status Code
Task Comments	Issue Comments

Table of Elements passed from MDCMS to Jira for Issues originating in MDCMS

MDCMS Element	Jira Element
Project ID	Project Key (must already exist in Jira)
Task Type	Issue Type
Task Status Code	Issue Status Code
Task Priority	Issue Priority
Task Due Date	Issue Due Date
Task Parent Task	Issue Parent Task
Task Assignee	Issue Assignee
Task Requester	Issue Creator
Task Summary	Issue Summary
Task Description	Issue Description

+ any fields that are mapped from MDCMS Custom Fields

How MDCMS Communicates with Jira

Jira Software and Jira Service Desk contain a large set of REST APIs that MDCMS can consume in order to retrieve or update information.

An MDCMS background service job named MDJIRA uses these REST APIs. It is an MDCMS program that polls for deltas to issues for applicable projects every n seconds or when triggered by a webhook from Jira or an update to be sent to Jira.

Only one partition needs to connect to Jira (typically the development partition). This partition should be set as a Workflow location and the MDPUSH/MDPULL jobs should be used to share the Jira information with the other locations.

The authorization of MDCMS to invoke the Jira APIs is enabled using either an API Token when using the Jira Cloud version or an Access Token when using the DataCenter/Server version. Support for authorization via OAuth 1.0 has been discontinued by Jira and MDCMS.

Prerequisites for using the Interface

- An active MDCMS and MDWorkflow license (v8+) on the IBM i partition used to connect to Jira
- The Portable App Solutions Environment IBM Licensed Program (SS1 option 33) must be installed
- Jira must be reachable within the network by the partition
- The Jira-Side configuration must be performed by a user with administrator access in Jira
- The MDCMS-Side configuration must be performed by a user with MDSEC authority to System Settings (code 11)

Configure the Interface

Jira Interface Settings

The interface settings are managed from MDCMS -> MDCMS Setup Menu -> Services -> MDJIRA

Common Service Settings

The common service settings are available on the initial MDJIRA service screen.

Property	Description
Auto-Start Jobs	<p>IF using the MDCMS WebHook for Jira, Auto-Start can be set to Y, since changes will be triggered by Jira.</p> <p>Otherwise, MDJIRA jobs can't be started automatically (except during the initial setup phase). Instead, command MDSTRJIRA should be added to the job scheduler.</p>
Default Job Queue	The name and library of the job queue that the MDJIRA1 and MDJIRA2 jobs should be submitted to. The default of QSYSNOMAX runs in subsystem QSYSWRK.
Default End Time	The time of day that running MDJIRA jobs should cleanly end *NEVER – the MDENDJIRA command will be used
Default Delay Interval	<p>The amount of time in seconds between checks for updates to projects and issues.</p> <p>If using the MDCMS WebHook for Jira, it is recommended to set the interval to 30 minutes (1800 seconds). Otherwise, it is recommended to set it to 30 seconds.</p>

Jira-Specific Settings

Press F10 from the MDCMS Service screen for MDJIRA to get to settings specific to the Jira interface.

Property	Description
Jira Base URL	<p>The domain name of the Jira web application for the https connection</p> <p>Example: https://my-company.atlassian.net</p>
API Token User	<p>The user ID (typically an email address) of the user that has generated the API Token in Jira Cloud for authorization to the Jira REST APIs.</p> <p>Leave blank if using Access Tokens in Jira Data Center/Server</p>
New API Token	<p>Press F14 to enter the value of the API Token for REST authentication.</p> <p>API Tokens are used when using the Jira Cloud version.</p> <p>For security reasons, a previously entered token value cannot be viewed. However, the Token Registered value will be set to API Token if a value is registered in MDCMS.</p>
New Access Token	<p>Press F15 to enter the value of the Access Token for REST authentication.</p> <p>Access Tokens are used when using the Jira Data Center/Server version.</p> <p>For security reasons, a previously entered token value cannot be viewed. However, the Token Registered value will be set to Access Token if a value is registered in MDCMS.</p>
Interface Enabled	<p>N – MDCMS will not try to communicate with Jira (unless Test Connection is used). This can be useful when making network or mapping changes to avoid traffic and data utilization when it should not occur.</p> <p>Y – MDCMS will allow for the mapping, import and export of information with Jira.</p>

Property	Description
Admin Time Offset	<p>Jira delivers timestamp information in the time zone of the Jira server. If the Jira administrator that authorized the connection is defined within Jira to be in a different time zone, then the difference between the server time zone and user time zone needs to be defined so that the query for issue differences functions accurately.</p> <p>For example, if the administrator is ahead of the server by 2 hours, then set Admin Time Offset to +0200.</p> <p>If the administrator is behind the server by 4 hours, then set Admin Time Offset to -0400.</p> <p>To troubleshoot the offset, make an update to an issue for a considered project and then review the Jira service log.</p> <p>If set too early, then the same deltas will continue to be listed in the pull response for n hours. If set too late, then changes won't be pulled into MDCMS until n hours later.</p>
Proxy Host	The address of the Proxy server to route http requests through
Proxy Port	The port number on the Proxy server to route http requests through
Server Version	<p>This helps ensure that the correct REST API syntax is used as Atlassian continues to evolve their API catalog.</p> <p>CLOUD - you are using Jira Cloud</p> <p>the server version in format vv.rr when you are using Jira Server or Jira Datacenter. For example, enter 07.12 when using Jira Server 7.12.nn</p> <p>You can find the version in Administration → System Support → System info</p>

Authorize MDCMS in Jira using an API Token

1. A Jira user with sufficient rights to the project(s) that will be relevant in MDCMS should log into Jira.
2. Click on the account icon at the top-right of the screen
3. Click on the Manage account option

4. Click the Security option in the Atlassian account panel on the left
5. Under the API token section, click on the option Create an manage API tokens
6. Click the Create API token button
7. Provide a label, such as MDCMS and then click the Create button
8. Copy the token string
9. MDCMS -> MDCMS Setup Menu -> Services -> MDJIRA and press F10 to view Jira specific settings
10. Ensure the Jira Base URL is set correctly, enter the user id of the user that generated the token (typically an email address) in the API User field and paste the token string in the New API Token field and press Enter.
11. Press F7=Authorize MDCMS in Jira. This will test the connection to Jira. If there is an issue, press F10=Logs to view details about the connection attempt.

Map Project/Task Settings

Before Projects and Issues can be shared between Jira and MDCMS, the Jira settings need to be mapped to MDCMS settings. These settings can be managed by pressing F8 from the Jira Interface Settings screen for the MDJIRA service.

Jira Projects -> MDCMS Projects

Option 1 from the mapping menu lists all Projects defined in Jira and allows you to specify what the equivalent MDCMS Project is for each Project that should be interfaced with MDCMS.

Press F5 to build or refresh the list of Jira Projects. The MDJIRA service must be activated to do so.

Press F9 to toggle the list between all entries and only mapped entries.

Use the filter fields to limit the list to rows containing the filter values.

Property	Description
Jira Project	The Project Key in Jira
Category	The Project Category in Jira, if defined, or *undefined if the Project isn't categorized
Description	The Project Description in Jira
MD Project	The MDCMS Project to map the Jira Project to If left blank, then the project will be omitted from transfer to/from MDCMS, which is a good method to avoid seeing irrelevant information in MDCMS. Press F4 on the field to manage and select a valid MDCMS Project

Property	Description
	Multiple Jira Projects can be mapped to the same MDCMS Project
O (Issue Origin)	Flag designating where issues for the given Project Category may originate J - the Issues for the project will only originate in Jira and be exported to MDCMS as Tasks M - the Tasks for the project will only originate in MDCMS (or somewhere other than Jira) and be exported to Jira as Issues B - the Project may contain issues that originate in either Jira or MDCMS

Jira Issue Types -> MDCMS Task Types

Option 2 from the mapping menu lists all Issue Types defined in Jira and allows you to specify what the equivalent MDCMS Task Type is.

Press F5 to build or refresh the list of Jira Issue Types. The MDJIRA service must be activated and running to do so.

Property	Description
Jira Issue Type	The Issue Type defined in Jira that can be attributed to Issues.
Description	The Description of the Jira element
Task Type	The MDCMS Task Type to map to If left blank, then all issues of the given type will be omitted from transfer to MDCMS, which is a good method to avoid seeing irrelevant information in MDCMS. Press F4 on the field to manage and select a valid MDCMS Task Type Multiple Jira Issue Types can be mapped to the same MDCMS Task Type
O (Issue Origin)	Flag designating where issues for the given Issue Type may originate J - the Issues of the given type will only originate in Jira and be exported to MDCMS as Tasks M - the Tasks of the given type will only originate in MDCMS (or

Property	Description
	somewhere other than Jira) and be exported to Jira as Issues B - the Issue Type may originate in either Jira or MDCMS For the import or export to Jira, the appropriate flags must also be set on the Task Type in the Task Type settings.
Task Only	Set this to Y if an issue of the given Issue Type should always be mapped as a task in MDCMS, even if the issue has a parent issue in Jira.

Jira Users <-> MDCMS Users

Option 3 from the mapping menu lists all Users defined in Jira and allows you to specify what the equivalent MDSEC User Profile is.

Press F5 to build or refresh the list of Jira Users. The MDJIRA service must be activated to do so.

Use the filter fields to limit the list to rows containing the filter values.

If the Jira user's display name or email address matches a MDSEC user description or email address, MDCMS will automatically suggest the MDSEC user profile and then you only need to press Enter to confirm.

You only need to map Jira users, if they will potentially be creating projects, creating issues, or be assigned to issues that are relevant to MDCMS

When a user mapping is confirmed, MDCMS will automatically store the Jira user's email address, if an email address isn't already defined for the MDSEC user.

Property	Description
Only Import Issue if Assigned to a mapped User?	Y - An issue will only be imported from Jira to MDCMS if the assignee for the issue is mapped to an MDCMS user N - An issue will be imported from Jira to MDCMS, even if not assigned, or if assigned to someone that isn't mapped. All other import rules continue to apply, though.

Property	Description
Jira User Name	The Jira login ID

Property	Description
Display Name	The Description of the Jira user
MDCMS User	The MDCMS user to map to. Press F4 to select from list of defined users If left blank, corresponding projects and issues will still be imported, but the user will be anonymous.
O (User Origin)	Flag designating the user to map based on the direction of the data flow (import or export), since there may be duplicate users in MDCMS or in Jira. J - when importing user-related data from Jira, map the given Jira user to the specified MDCMS user. Multiple Jira users can map to the same MDCMS user M - when exporting user-related data to Jira, map the specified MDCMS user to the given Jira user. The MDCMS user can only be mapped once for export. B - when importing or exporting user-related data, map the specified MDCMS user to the given Jira user. The MDCMS user can only be mapped once for export.

Jira Issue Fields <-> MDCMS Custom Fields

Option 4 from the mapping menu lists all Fields defined for use in Jira issues and allows you to specify what the equivalent Task Custom Field is.

Fields that auto-map to primary Task fields in MDCMS aren't included in the list.

Press F5 to build or refresh the list of Jira Fields. The MDJIRA service must be activated to do so.

Press F9 to toggle the list between all entries and only mapped entries.

Use the filter fields to limit the list to rows containing the filter values.

Property	Description
Jira Field	The Jira field name
Type	The type of field in Jira, to aid in selecting the appropriate Custom Field in MDCMS. array(<type>) indicates that the field is a list of the given type.

Property	Description
MDCMS Field	<p>The MDCMS field to map to. Press F4 to select from list of defined fields</p> <p>If left blank, corresponding issues will still be imported, but the field won't be included in the import or export.</p>
O (Field Origin)	<p>Flag designating the direction of the data flow (import or export) for the field.</p> <p>J - only import the field value from Jira M - only export the field value to Jira B - pass the field value in both directions, indicating that editing of the field is intended in either product.</p>

Jira Status Transition -> MDCMS Status Transition

Option 6 from the mapping menu lists all Status Codes defined in Jira and allows you to specify what the equivalent MDCMS Status Code is in order to automate the transition of a status for an MDCMS Task when the issue status transition occurs in Jira..

Press F5 to build or refresh the list of Jira Status Codes. The MDJIRA service must be activated to do so. Press F9 to toggle the list between all entries and only mapped entries.

Property	Description
Jira Status	The Status Code defined in Jira that can be attributed to Issues.
Description	The Description of the Jira element
MD Status	<p>The MDCMS Status Code to map to</p> <p>If left blank, then all issues currently in that status will not cause the creation or update of the task in MDCMS. This is a good method to allow the preliminary request and review process to remain entirely in Jira and then flow into MDCMS once work should occur.</p>

Property	Description
	<p>Press F4 on the field to manage and select a valid MDCMS Status Code</p> <p>Multiple Jira Status Codes can be mapped to the same MDCMS Status Code</p>
Update Only	<p>Y – update the information for the task in MDCMS when the Jira issue is in that status, but don't create the task in MDCMS when it doesn't exist yet. This is to update MDCMS when Jira transitions back to an earlier status that normally wouldn't flow into MDCMS or to avoid bringing over Jira history for closed issues during the initial import.</p>
O (Issue Origin)	<p>Flag that limits the status transitioning based on where the Issue originated.</p> <p>J - only perform the status transition in MDCMS if the issue originated in Jira</p> <p>M - only perform the status transition in MDCMS if the issue originated in MDCMS (or anywhere except Jira)</p> <p>B - perform the status transition regardless of where the issue was created.</p>
Exceptions	<p>The Status Transition that occurs in Jira can be ignored in MDCMS when the current status in MDCMS is defined on the exception list for the new Status value.</p> <p>To list/define status exceptions, position the cursor to the row and press F8. Place a Y on each row when the update in MDCMS to the new Status should now occur if the MDCMS status has the row's status. This is helpful if you have additional transitions within MDCMS for better granularity for your dev team, so that the MDCMS status won't revert to a code that is less granular.</p>

MDCMS Status Transition -> Jira Status Transition

Option 7 from the mapping menu lists all Status Codes defined in MDCMS and allows you to specify which of those codes are allowed to be updated (transitioned) to by specifying the Jira Status that MDCMS will request to be transitioned to. If the issue originated in MDCMS, a status transition in MDCMS will be allowed even if it isn't mapped to a Jira Status.

Press F5 to build or refresh the list of Jira Status Codes. The MDJIRA service must be activated to do so.
Press F9 to toggle the list between all entries and only mapped entries.

Property	Description
MDCMS Status	The Status Code defined in MDCMS that is allowed to be transitioned to within MDCMS for Jira issues
Jira Status	<p>The name of the Jira Status Code that will be transitioned to within Jira. Press F4 to select from list.</p> <p>The special value of *internal can be used to allow the transition with MDCMS, but not to trigger a transition within Jira.</p> <p>This is helpful if you have additional transitions within MDCMS for better granularity for your dev team.</p>

IMPORTANT - Prerequisites for the Jira Status Update to function:

- the Workflow that is defined in Jira for the Issue's type must allow the transition from the current status in Jira to the status that is mapped from the MDCMS status
- the IBM i partition that connects to Jira must be defined in the OS/400 Locations on the partition initiating the status change. If it's the same system, then the Location ID is *LOCAL
- the OS/400 Location definition must have the MDWorkflow Rep parameter set to Y and assigned either a Push Job# or Pull Job#
- the MDPUSH or MDPULL service must be running
- the MDJIRA service must be running

Configure MDCMS WebHook in Jira

Option 9 from the mapping menu allows you to create or update the MDCMS WebHook in Jira. The WebHook is used as a trigger whenever an issue of a mapped issue type for a project of a mapped project category is created or updated. There are 2 significant benefits to using WebHooks:

- No delay for importing new or updated issues from Jira
- the delay interval between checks for changes on the Jira server can be reduced from seconds to a half-hour or more to limit network traffic, processing use and log size.

Before the MDCMS WebHook is registered in Jira, the MDCMS REST API server needs to running.

Press F6 to add the WebHook. MDCMS will then automatically register the WebHook on the Jira server with the appropriate conditions to limit traffic to relevant issues.

Once the WebHook is created, it should then be updated anytime the mapping changes for Project Categories or Issue Types.

If the WebHook should no longer exist on the server, use F22 to delete it.

If you need to change the URL of the MDCMS server, you should first delete the WebHook, then rename the URL and finally add the WebHook again.

Pull Issues from Jira

Full Import

Once the mapping is complete for projects, issue types, users, fields and status codes, F9=Get Issues can be pressed from the Jira Interface Settings for the MDJIRA service.

You are then prompted for what should be imported.

If Jira Project is set to *ALL, then issues for all mapped Jira projects will be imported. Otherwise, enter the key of a specific Jira project to only retrieve issues for it.

If From Date is set to *ALL, then all issues will be imported regardless of the age of the issue. Otherwise, enter a specific From Date and From Time to only import issues that have been created or modified since the entered date/time.

A full import should be performed initially and whenever an Issue Type or Status Transition is added to the mapping.

Delta Import

Once an initial Full Import is performed, the MDJIRA service will check every n seconds for any new or changed issues since the last check for each Jira Project that is registered in MDCMS. The bandwidth for this request is very small since only differences from the prior check are returned.

If using the MDCMS REST Webhook, it's recommended to set the interval at 30 minutes (1800 seconds).

If not using the Webhook, it's recommended to set the interval to 30 seconds. Changes are instantaneous and bandwidth is minimized when using the Webhook, so the use of the Webhook is very highly recommended.

Registered Jira Projects will continue to be queried until they are closed in MDCMS.

Push Comments to Jira

Include Comment when MDCMS Triggers Transition

When a mapped status changes in MDCMS, the MDJIRA service will attempt to transition to the mapped Jira Status in Jira and comment that it was performed by MDCMS.

The comment about the status change will only be stored with the transition in Jira if the transition is attached to a screen.

If you wish to have the comment saved with the transition, and don't otherwise have screens already defined for the transitions, do the following in Jira:

1. Navigate to Settings->Issues->Screens
2. If an empty screen doesn't exist, add a new screen, name it Comment Screen and click Add
3. Navigate to Settings->Issues->Workflows
4. Use action Edit for a Workflow that is used for Issues used in MDCMS
5. For each Transition in the Workflow that can be triggered from MDCMS, left-click on the Transition and then click Edit. A dialog will pop up where the Comment Screen can be selected and then click Save
6. Repeat for any other Workflows

Push Other Comments to Jira

Command MDCMS(ENV)/MDJIRACMT can be used to push information in the form of comments to Jira for either a specific Issue or for all impacted Jira Issues in an RFP. It is intended to let the Jira participants know about RFP activity in MDCMS and can be added as a *RFP command to occur at necessary exit points, such as after an install so that testing will proceed or when approval is required for an installation.

MDJIRACMT transactions are logged in file MDCMS(ENV)/MDDJIRR.

The MDCMS, MDXREF and MDSEC libraries must be in the library list when the command is invoked.

MDJIRCMT Parameters

Property	Description
Comment (CMT)	The comment to be added. The comment can be up to 800 characters in length
Ref Code (IREF)	The task reference code for the MDCMS task that matches the Jira issue key. *RFP (default value) – the comment will be added for each Jira issue that is impacted by the RFP. Include the AGP and RFP parameters for this. If the command is used from an RFP exit point, set AGP to ##APPLIC## and RFP to ##RFPNBR##.

Property	Description
	If the command used from an Object exit point, set IREF to <code>##TSKREF##</code> , including the single quotes.
Application (AGP)	The Application targeted by the RFP
RFP Number (RFP)	The RFP Number whose impacted Jira issues require a comment

View Jira Information from MDCMS

Jira Projects

The Jira Project Key is used as the Project ID in MDCMS, so searching and listing is the same as for any other project.

There is no limitation to what can be modified directly within MDCMS for Jira Projects.

Jira Issues

The Jira Issue Key is stored in the MDCMS Task Reference Code. The code value can be searched and listed in MDCMS by pressing F8=Ref Code in the Task Listing.

Only the Due Time, Test Group and Test User can be edited in MDCMS. All other fields must be edited directly in Jira. The Jira Issue transitions can only be performed within MDCMS for mapped statuses.

View Jira Information from MDOpen/MDWorkflow

Jira Projects

The Jira Project Key is used as the Project ID in MDOpen/MDWorkflow, so searching and listing is the same as for any other project.

There is no limitation to what can be modified directly within MDOpen/MDWorkflow for Jira Projects.

You can navigate directly to Jira from a Project in MDOpen/MDWorkflow by clicking on the URL link for a row in the Project Listing. For MDOpen to link to Jira instead of MDWorkflow, set URL for MDOpen to N in the Email Settings.

Jira Issues

The Jira Issue Key is stored in the Task Reference Code.

For MDOpen, the label for this code can be changed in the MDOpen System Settings by changing the value for field Task Reference Label.

For MDWorkflow, the label for this code can be changed in the conf-faces-config.xml configuration file by changing value for property iref.

Only the Due Time, Test Group, Test User, Application, Location, Level and Custom Fields can be edited in MDOpen/MDWorkflow. All other fields must be edited directly in Jira. The Jira Issue transitions can only be performed within MDOpen/MDWorkflow for mapped statuses.

You can navigate directly to Jira from a Task in MDOpen/MDWorkflow by clicking on the URL link for a row in the Task Listing.

What to do in MDCMS after Migrating Jira to a new Server

If the MDJIRA interface in MDCMS has actively been used, and the Jira server is migrated from one server to another, or migrated from Cloud to Server, or Server to Cloud, then it's possible that the Application Link will need to be reestablished and that the Jira indexes will have changed.

If transferring Jira Server from one server to another for the same version of Jira, it is highly recommended to NOT use the Jira XML import/export feature, because all elements will get new index numbers which can not only effect MDCMS, but potentially also other interfaces. Instead, export the entire database, such as PostgreSQL. This will keep all key index values intact.

If transferring between Server and Cloud, or switching to a new SQL product or version during the transfer, then the XML import/export will need to be used. In this case, perform the following steps to avoid losing any mapping definitions and minimizing interface downtime:

1. Backup or export the entire mapping table MDCMS/MDDJIRM

Projects (KEYFLD = PJ)

Project Categories (KEYFLD = PC)

Issue Types (KEYFLD = IT)

Status Codes (KEYFLD = ST)

are the mapping elements with a numeric index, which can change

2. Delete the Project records in table MDCMS/MDDJIRM (KEYFLD = PJ)

3. Note or make a print screen of the values in MDCMS Task Status Update → Jira Issue Status Update

4. Use F8=Mapping from the Jira Interface Settings and Press F5 for each of the mapping options (except Users) to get the new indexes.

If unable to connect, then check the address and verify that the Application Link is enabled. If no longer enabled, then Re-Authorize MDCMS in Jira.

Verify and update the elements to match the mapping in your backup/export of MDDJIRM

5. Use F9 to perform a Full Import. This will pull in the Project indexes and refresh the issue/task information.