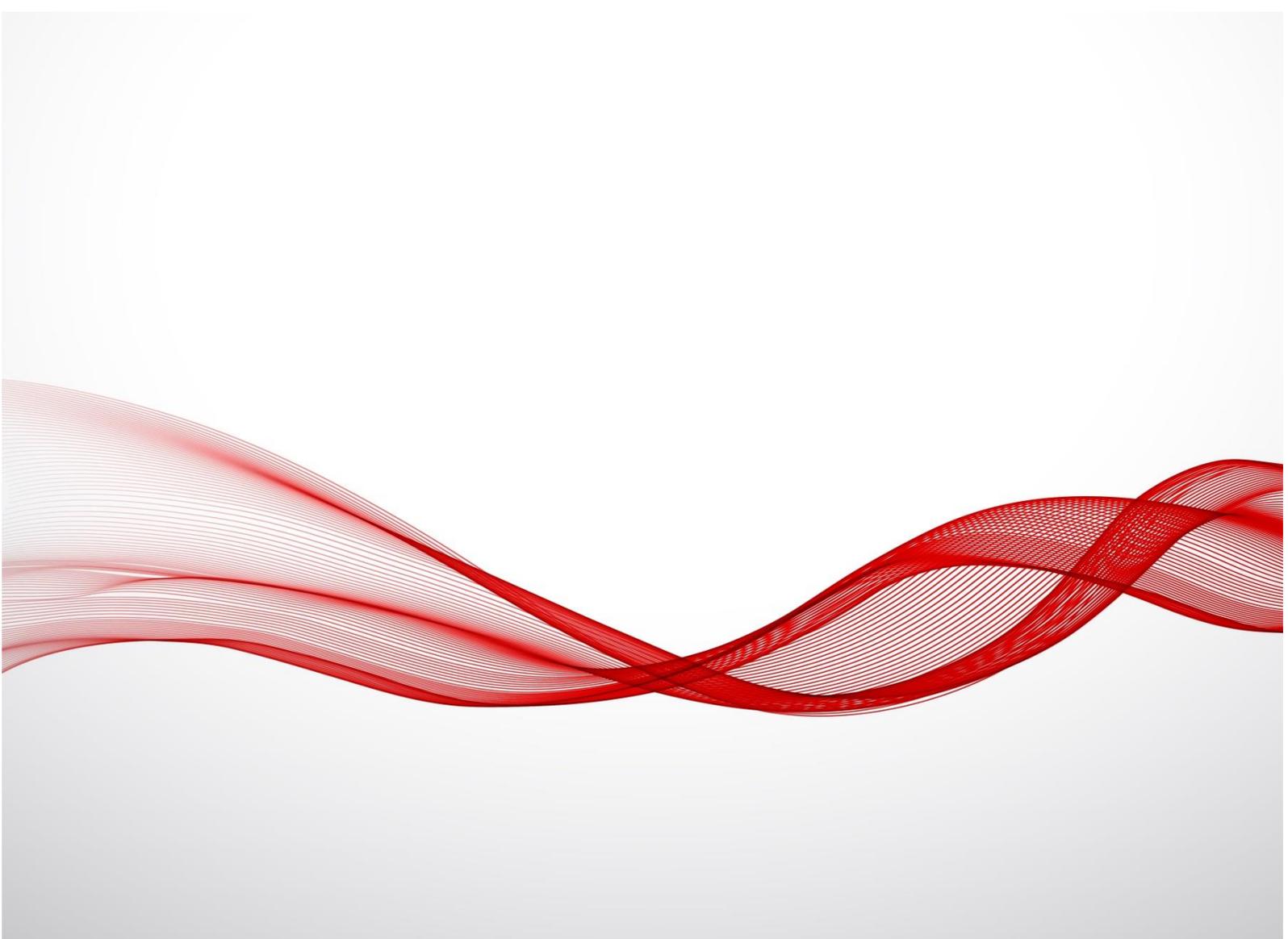


White Paper

Aramea RAQL

Product Overview



August 2016

Contacting Aramea

The Aramea website is <http://www.arameasoftware.com>

Email us at info@aramea.co or sales@aramea.co

For support, email support@aramea.co

Product documentation is available at <http://docs.aramea.co>

You can call Aramea on the following numbers:

USA: +1 321-710-7201

UK: +44 203-514-3079

Contents

Contacting Aramea	2
Preface	4
What is RAQL?	4
Problems interacting with Remedy data	4
Why RAQL?	4
Aramea RAQL.....	5
Utilising	5
Commands	6
Entering queries through Mid-tier	6
Aramea RAQL Client.....	6
Using RAQL from within workflow	6
Remedy Integration	7
Examples of RAQL Statements	7
Batch resolve Incidents	7
Batch re-assign Incidents	8
Create Sites for a company based on another	8
Copy support group associations	9
Creation of a new Task within a Change Request	9
Extracting all attachments from Incident Work Logs	9

Preface

What is RAQL?

RAQL is the Remedy Advanced Query Language. At its core it is a Java library providing a high-level SQL-like language to interact with data inside a BMC Remedy AR Server. On top of this, a front-end interface is provided to interact with your Remedy data through a Windows client or mid-tier, or through filter workflow.

Problems interacting with Remedy data

Traditionally interaction with Remedy data is through:

Workflow	Limited abilities. For example, you cannot retrieve distinct values, use group by or having clauses and joins require development of join forms which is inefficient for ad-hoc queries.
Remedy API	Requires specific knowledge and expertise and can take long to develop
Web Services	Difficult to define for more than simple queries and performance can be a problem.
SQL	Flexible but bypasses the Remedy Security Model
ODBC	The BMC ODBC Driver is no longer supported since Remedy 8.x and later
JDBC	Limited to Read-only operations and not officially supported
REST API	Limited to AR Server 9.0 or later communicating over HTTP, requires knowledge of and creation of JSON format files.

Why RAQL?

Aramea RAQL provides the flexibility of SQL but by parsing and then executing statements through the Remedy API, the Remedy security model is still maintained. Data can be queried, joining data from different Remedy forms ad-hoc without the need to create join forms. Data can also be created, updated or deleted. All statements trigger the appropriate filter workflow on the Remedy server when executed.

RAQL is being used by consultants to diagnose problems and speed up maintenance of Remedy data. Development teams are benefitting from faster development times for integration projects.

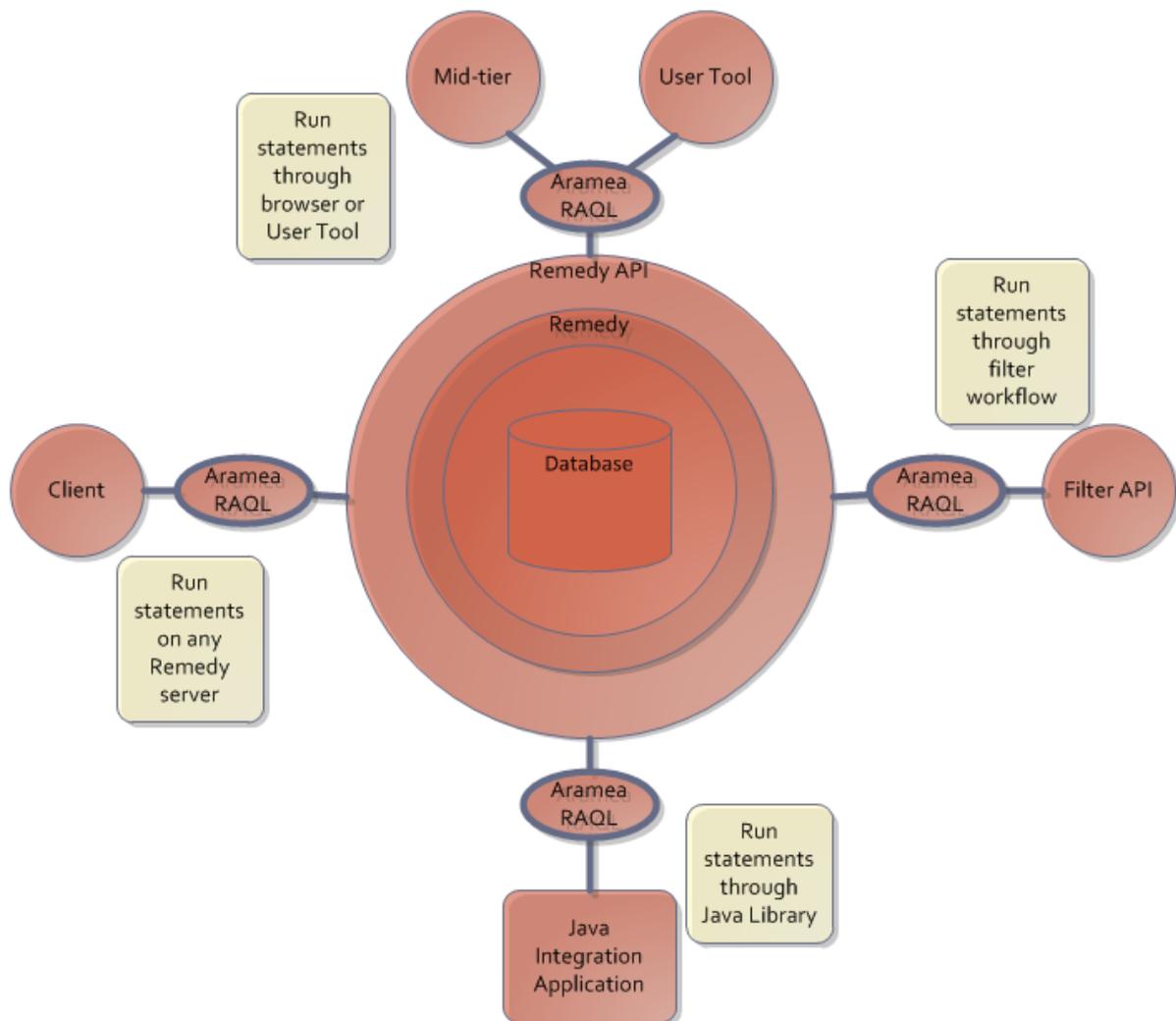
Aramea RAQL

Utilising

Aramea RAQL provides many ways in which to utilize the power and flexibility of the core product.

With the RAQL server software installed, you can enter statements in the front-end through Mid-tier or User Tool or you can use within your filter workflow as an alternative to the SQL Set Fields action.

The RAQL client software allows you to enter statements while connected to any Remedy server and the RAQL Java library greatly streamlines your integration projects.



Commands

RAQL utilizes 4 primary commands to allow you to manipulate the data in your BMC Remedy server.

You can retrieve information from your Remedy server using the **SELECT** statement.

If you wish to create new entries, use the **INSERT** statement.

Modify existing data by using the **UPDATE** statement.

Remove data from your Remedy server using the **DELETE** statement.

Note, if you login with a Remedy login that has a Read-only licence then you will only be able to run the **SELECT** command. RAQL observes the Remedy security model so you will only be able to access the field data, records and/or forms that your login is authorized.

Entering queries through Mid-tier

The Aramea RAQL server software provides a front-end through which you can enter statements from within Mid-tier (or the Remedy User Tool).

Results from the statement are displayed in table format, including any records created, updated or deleted.

You can dynamically add tabs to facilitate multiple statements, each with their own results table.

Aramea RAQL Client

The Aramea RAQL client software allows you to run RAQL statements on any Remedy server without the need to install any additional software on the server.

This can be a huge benefit to Remedy Developers, Administrators and professional services teams on sites where direct access to the database to run queries is not available.

Using RAQL from within workflow

The Aramea RAQL Filter API plugin allows you to run RAQL statements from within your filter workflow, as an alternative to running a Set Fields action with SQL directly on the database.

Remedy Integration

The Aramea RAQL Java library enables faster development of integration with your Remedy servers. Two lines of code is all that is required; one to create the session, the next to run the RAQL statement that returns the results.

This can dramatically reduce the time and cost of developing software for integration.

Examples of RAQL Statements

The power and flexibility of Aramea RAQL allows for many uses, whether you need to simply replace your SQL statements to protect data or you have an integration project.

Here are some examples of statements to highlight some of what can be achieved using RAQL, many of which have been used on customer sites to diagnose or resolve issues. Therefore some customer data and/or fields have been replaced.

Batch resolve Incidents

RAQL supports the ability to modify multiple records with a simple UPDATE statement. As shown in this example you can also modify hidden fields:

```
UPDATE "HPD:Help Desk"  
SET Status = 'Resolved',  
Assignee = 'Chris Jones',  
"Assignee Login ID" = 'CJONES',  
Resolution = 'Closed by Chris Jones',  
"Resolution Method" = 'Cancellation',  
"Actual Resolution Date" = $TIMESTAMP$,  
"ResolutionCategorization Tier 1" = 'Tier1',  
"ResolutionCategorization Tier 2" = 'Tier2',  
"ResolutionCategorization Tier 3" = 'Tier3',  
Status_Reason = 'No Further Action Required',  
"SLA Responded" = 'No',  
CurrentStageNumber = 3  
WHERE (Status != 'Resolved')  
AND ("Incident Number" IN ('INC000002285042', 'INC000002160479', 'INC000002217139',  
'INC000002227210', 'INC000002299560'))
```

The IN command supports the listing of all incident numbers to be resolved.

Batch re-assign Incidents

After a new support group was created we needed to re-assign all open tickets from the old support group that was being replaced, so this UPDATE statement allowed us to do that:

```
UPDATE "HPD:Help Desk"  
SET "Assigned Group ID" = 'SGP000000013334', "Assigned Support Company" = 'Company',  
"Assigned Support Organization" = 'Organization', "Assigned Group" = 'Support Group'  
where "Assigned Group ID" = 'SGP000000002558' and Status < 'Resolved'
```

Create Sites for a company based on another

We needed to copy all the configured sites for one company to another. This was done through the following two statements. Note that this customer has the Company field added to the site form.

1. Create sites for one company based on the list for another

```
INSERT INTO SIT:Site(Company, Site, Street, Country, "State Province", City, "Zip/Postal Code", "Time Zone")  
VALUES  
(SELECT 'Company B', Site, Street, Country, "State Province", City, "Zip/Postal Code", "Time Zone"  
FROM SIT:Site WHERE Company = 'Company A' AND Site NOT IN  
(SELECT Site FROM SIT:Site WHERE Company = 'Company A'))
```

2. Create the Site Company Associations

```
INSERT INTO "SIT:Site Company Association"("Site ID", Company)  
VALUES  
(SELECT "Site ID", 'Company B' FROM "SIT:Site Company Association"  
WHERE Company = 'Company A'  
AND "Site ID" NOT IN (SELECT "Site ID" FROM "SIT:Site Company Association" WHERE Company = 'Company B'))
```

Data is created using the INSERT statement using either literal values or from the results of a SELECT statement. Note the above statements also make use of the NOT IN with a sub query to prevent duplicate data created.

Copy support group associations

We often have requests to add users to the same support groups as another user that are not specified in their people template. This example shows how this can be done quickly and easily using RAQL instead of manually performing the same task:

```
insert into "CTM:Support Group Association"  
("Login ID", "Person ID", "Full Name", "Assignment Availability", "Support Group ID", "Support Group  
Association Role", Default)  
values  
(select 'CJONES', 'PPL000001016024', 'Chris Jones', "Assignment Availability", "Support Group ID", "Support  
Group Association Role", Default  
from "CTM:Support Group Association"  
where "Login ID" = 'XYZ')
```

Creation of a new Task within a Change Request

This example shows that by using RAQL you can create a task for a Change Request with a single statement by using information from the Task Template and the Change itself joined to provide the data for the INSERT statement

```
INSERT INTO TMS:Task  
(TemplateID, TaskName, Summary, "Location Company", RootRequestMode, RootRequestName,  
RootRequestInstanceID)  
VALUES  
(SELECT T.InstanceId, T.TaskName, T.Summary, 'Calbro Services', 'Real',  
C."Infrastructure Change ID", C.InstanceId  
FROM TMS:TaskTemplate T, "CHG:Infrastructure Change" C  
WHERE T.Summary = 'Standard Update of CMDB'  
AND C."Infrastructure Change ID" = 'CRQ000000000001')
```

As this is not SQL but RAQL the relevant workflow will be triggered for Submit filters when the statement is executed.

Extracting all attachments from Incident Work Logs

This example shows the use of a special RAQL function used to extract all the attachments

```
SELECT TO_FILE("z2AF Work Log", 'C:\temp\attachments') FROM HPD:WorkLog
```

Please contact Aramea if you have any specific questions on how RAQL can be used to assist your requirements.