Remedy AR System API and Integration Interfaces Overview (c-api, java, dotnet, web-services, email, ruby, jython, vb, direct-sql)

<< Other AR Resources

Topic

This document is intended to be a landing pad for **all** the application programming interfaces available to developers.

Here is the Integration Guide for 7.6.04 from BMC.

The Client Type APIs

There are several programming languages available for building Remedy clients.

Both the native Windows client (aruser), Mid-Tier and Developer Studio has been built using this type of API, along with most third party integrations to the AR System.

If you are building your own Developer Studio, you will need to stick to Java or C, but in 99% of the cases you are only working with Data, and in those cases all the interfaces are ok.

If you only need to create, search or update data within the AR System, you can let your environment and programming skills direct your choise of language.

Technology	Pros	Cons	Quick Start / Comparison	Download	Learning more
C	• Can do everything	• Needs to be compiled by for	C	C	

		• each platforms • It is kind of hard to code C			
Java	• Can do everything • Platform independent • Focus of BMC to run more and more in Java		Java	Java	Java docs: 7.6.4 7.5 7.1 7.0
Driver Program	• "Can do everything" • Precompiled by BMC	• Not good at error handling • Command line interface only	Driver	Driver	
Perl	• Platform independent • Interpreted language • Ease of use	• Not supported by BMC • Can be hard to compile for your platform	Perl	ARSperl	arsperl.org
.Net	• Easy to code if you are a Microsoft person	• Not supported by BMC • Windows platform only	C# VB	.Net/COM: 7.6.4 7.1 7.0	.Net/COM: part 1 part 2
Ruby	• Next generation web language • Rides on top of Java via JRuby	• Not supported by BMC, however uses Remedy Java API	Ruby		Ruby JRuby Remedy Javadocs API

	• All the benefits of Java • All the benefits of Ruby				
Jython	• Directly imports AR System Java API Classes as well as any other Java Class you may require • All the benefits of a dynamic language coupled with the power of Java and the AR System Java API • Platform independant	• Not supported by BMC, however uses Remedy Java API	Jython		Remedy Javadocs API Reference Jython
PHP		• Not supported by BMC			
Python		• Not supported by BMC			

The Filter Plugin API for extending Filter Actions to other programming languages

This interface is called from filter actions, and it allows you to send/recieve field data to an external program, either C and Java.

This is useful for building integration that should trigger based on client activity, such as saving or updating a record.

The ARDBC Plugin API for integrating other Database Type resources into the AR System

This API is used when you want to make an external data source available for Users and/or Remedy Developers in the same way as any form. They are called Vendor Forms.

Your integration can allow searching, viewing, updating and creation of data in the external data source.

A pre-build integration for LDAP (AD) using ARDBC technology is shipped with the AR Server.

The AREA Plugin API for authentication integrations

This plugin is used when you keep your passwords, and/or account information such as email, license type and permission groups outside your AR System database.

An LDAP integration built on this API is shipped with the system to integrate to, for example, Active Directory (AD).

Consuming external Web Services from the AR System

You can call external web services from filters, in order to push or get data into the AR System.

This is done via a Plugin that is shipped with the AR Server, and runs within the Plugin server.

- Web Services Best Practices

Publishing Web Services from the AR System

Publishing data or interfaces to the AR System can be done on Mid-Tier, via a HTTP Web Service interface.

This is useful when you want other systems to submit, search or modify data within your AR Server.

- Primer on Publishing Web Services
- Using BMC ITSM OOB Web Services
- Web Services Best Practices

Data Visualization Field integrations

You can seamlessly incorporate other web content, or your own rich web applications within a Data Visualization field, to allow better user interaction.

The programmer can hook into the user session in order to get or set data in the users session, or to retrieve/update additional information directly from the AR Server. The latter is done via the ordinary Client Type Java API described above, while leveraging the users session to the server, including permissions and the AR License.

This is the Web-replacement for client side Run Process or COM integrations executed from within the ARUser Windows client.

Integration through Email Engine

You can use emails to send notifications from the AR System, or to receive emails from users or other system.

You can send email commands to the AR System in order to submit, modify or search for tickets using both a native format and an XML-based format.

Direct SQL integrations

You can have your AR System issue SQL-commands within the database it is connected to.

The SQL commands can be executed from Filters, Active Links or Character Menus (menus connected to individual fields).

You can access external databases as well, but these has to be made available within the (single) database that your AR Server connects to. For example via a database link.

Database integration using View Forms

If you need flexible integrations to a database table, you can point a View Form to this table.

This would typically be a table or view containing data external to the AR System, but it must be data that can be accessed from the (single) database that your AR Server connects to. For example via a database link.

BMC Atrium Integration Engine (AIE/EIE)

The Atrium Integration Engine is a mechanism to manage data transfers between external data sources and the CMDB along with AR System forms. It is a bi-directional and multi-threaded data transfer engine. It supports event based or scheduled information flow. It is ideal for bulk, incremental data syncronization transfers.

AIE provides an intuitive graphical interface for field mapping and job configurations.

Data sources include:

- Flat file e.g. csv, XML
- Database using the associated connectors - SQL, Oracle and DB2

- Custom built adapters via Adapter Development Kit

With BMC Atrium Integration Engine, you can do scheduled bulk data transfers and event-based integrations initiated by either the source, target, or any other application. You can also use BMC Atrium Integration Engine for initial data load, incremental data transfers, and data synchronization. You can build links between BMC Remedy IT Service Management (BMC Remedy ITSM) applications and Enterprise Resource Planning (ERP), Customer Relationship Management, Supply Chain Management, and other enterprise applications.

BMC Atrium Integrator (7.6.x and above)

Atrium Integrator is an integration engine that enables you to transfer data from external data stores to BMC Atrium Configuration Management Database (BMC Atrium CMDB) classes. You can connect to a wide variety of input sources using connectors such as JDBC, ODBC, JMS, native databases, web services, and complex XML. Atrium Integrator provides you with the ability to clean and transform your data before putting it into BMC Atrium CMDB. Atrium Integrator replaces BMC Atrium Integration Engine. You can continue using BMC Atrium Integration Engine for existing data mappings and exchanges, but BMC recommends that you use Atrium Integrator for all new data transfers. BMC Atrium Integration Engine will be deprecated in a future release.

Remedy AR System API and Integration Interfaces Overview (c-api, java, dotnet, web-services, email, ruby, jython, vb, direct-sql)