OMDB: CmdBuild @ Maximus

Barry Leibson, Performance Engineering, Maximus

1. What does Maximus do and for whom?
2. What does my team do and how do I fit in?
3. What is OMDB?
4. Why we chose CmdBuild Ready2Use for OMDB
5. The ways we use CMDBuild that might be atypical
6. Q & A

maximus

What does Maximus do and for whom?

- Maximus is a large (thousands of employees, dozens of offices) U.S.-based company that provides services to government agencies. Most of our business is in the U.S., often at the state level, but Maximus now provides services to the governments of several other countries.
- Currently, most of the work falls into one of two areas: health* and human resources. For example, Maximus helped the U.S. government put together a unified response to the COVID-19 epidemic.
- Maximus operates many call-centers where people can get their questions answered. We also provide self-service web-based solutions and informational sites.

*Those outside the U.S. may have trouble believing how patchwork and complicated the health care system is. Americans need lots of help buying insurance and navigating state and federal programs. Maximus plays a significant role there.

maximus

What does my team do and how do I fit in?

- My team, Performance Engineering (PE), is part of Maximus IT, a very large organization within Maximus.
- PE collects IT data and helps the IT staff make effective use of that data. For example, we generate alerts when a server goes down unexpectedly or when a network device stops routing traffic. We also provide dashboards and reports for IT staff and leaders. We report on servers running end-of-life software to help prioritize upgrades.
- Among the software packages we use to collect and display data: Splunk, SolarWinds, AppDynamics, and CmdBuild ReadyToUse
- As a member of PE, every day is different, but often I'm busy
  - enhancing and running our CmdBuild site (which of course you'll be hearing more about).
  - moving data from here to there, often transforming it along the way.

Every couple years or so, my wife and I vacation in Italy. We usually spend a week in Lucca and a week in another location nearby (in Toscana or Umbria)

maximus

What is OMDB?

- OMDB is an acronym for Operations Management DataBase, a name my old boss's boss invented
- It's a subnet of what you might find in a traditional CMDB.
- It replaced an earlier home-grown system used by our application administrators to track what servers were running what and for whom.

Why we chose CmdBuild Ready2Use for OMDB

- Since OMDB is a subset of CMDB, it made sense to use a CMDB solution.
- Because we needed to duplicate functionality built into the system that we were replacing (that home-grown one), we needed a system that was customizable.
- Because we expected to use automated processes to import and export data from other systems on a regular basis, we needed a system that made that easy
- Because we expected people with differing roles within Maximus to add data and form relationships among the data, we needed a good user interface and a sophisticated access-control system (RBAC)
- We didn't have a large budget.

In the end, CMDBuild easily met all our criteria, and we liked the Tecnoteca people we met while we were evaluating it.

The only thing we got wrong in our evaluation was the effort involved in using connectors to exchange data – more on that later.

maximus

The ways we use CMDBuild that might be atypical

- While some of the data is maintained by OMDB users logging in and modifying the data, most of the data arrives via automated processes.
- We have defined many classes and augmented many of the others. We love and appreciate the CMDBuild's class implementation, especially the way inheritance works.
- We don't use most the of CMDBuild Ready2Use modules, such as Incident Management, Request Fulfilment, and Change Management.
- Our Field Services team has built a web-based app that treats OMDB as its data-store

Pulling all that data in

- When we discovered that building connectors involved Java (Groovy) programming and were single purpose, we shifted to importing CSVs via the Import/Export mechanism, which is quite intelligent (lots of merge options and detailed error reporting)
- As I'm a scripter (Bash and Python primarily), I started writing scripts that generally follow this recipe:
  1) Request data from a system via its API
  2) Parse the JSON or XML that comes back, isolating the information I'm looking for
  3) Write one or more CSVs. Often, more than one as I'm not only modifying a class, I'm creating relationships
  4) Stage the CSVs in our "Import" directory

- We now employ dozens of these such scripts that pull data from several systems, mostly on once-a-night schedule, and our CMDBuild Task Manager has about 75 scheduled imports now.

NOTE: Because we have two instances of CMDBuild running behind a load balancer and couldn't know which one would execute an import task, we needed a shared Import directory. We solved this by sharing a directory housed on server1 with server2 via NFS.

maximus