



University of California
San Francisco

Platform as a Service (PaaS)

Policies and Procedures



PaaS Policies and Procedures

- Purpose of this document is to define what a Platform as a Service (PaaS) customer will need to do in order to use the Mule platform (Enterprise Service Bus and API Manager)
- Policies and Procedures
 - Platform Access
 - Integrated Development Environment (IDE) Setup
 - Code Deployments
 - Patches and Upgrades
 - Unplanned Events
 - Coding and Design Standards
 - More to be determined...

Objectives

- Create an enterprise platform for other teams to use.
- Ensure a stable and reliable platform for all interfaces.
- Foster integration best practices and educate others on integration.

Platform Access Requirements

- We want our users to be successful with the platform and tool.

Therefore, we need to gauge the user's knowledge and experience with integration practices and technologies.

- Users wishing to access the Mule platform will need to:
 - Request a PaaS initial consultation (required meeting)
 - This is a short meeting between the users and the Integration Services team
 - Share past experience(s) in using these tools or building integrations
 - Users without integration experience will need obtain [Mulesoft's online certification](#)

Platform Access

- Platform Access - The Integration Services team will need to create your team's accounts and grant access to the platform.
 - Customer should first read the PaaS Policies (this presentation)
 - Customer should next read [MuleSoft Environment Access for Developers](#) and determine what access is needed
 - Customer will need to create a ServiceNow ticket for the Integration Services team (assignment group ITS_BA_Integration_Services) requesting access to the Mule Platform
 - Indicate which environments, names, and roles for each user
 - Interface team will work with the server team to setup users in a timely manner

IDE Setup

- The Integration Services team will provide guides to help your team get the tools installed and configured
 - Customers should read [Getting Started](#)
 - *The Getting Started guide is a living document and will be updated as we learn what customers are asking and need to know.*

Code Deployments

- DEV and local deployments can be done by the customer as needed
- The Integration Services team will need to deploy your code to STAGE and PROD.
 - Customers will need to open a Service Now ticket <https://ucsf.service-now.com/> for assignment group ITS_BA_Integration_Services
 - **Users will need to include business justification needs if any deployment on holidays, weekend or outside of regular business hours (8am-5pm, M-F).**
- STAGE deployments will require:
 - Interface Specification (use format provided by Integration Services team)
 - Design Diagram(s)
 - Code Review
- PROD deployments will require:
 - Completion of UCSF ITS Security Review process
 - Completion of PROD checklist (will be provided by Integration Services team)
 - Change Advisory Board (CAB) Approval – customer to present and request to CAB
 - Go Live Implementation, Deployment, Communications and Backout Plans
 - Performance and Functional Test Results

Patches and Upgrades

All interface owners must plan and prepare for patching and upgrading

- Patches – The Integration Services team will need to patch the environments to support planned and unplanned events
- Upgrades – The Integration Services team will need to upgrade the environments 2 to 3 times per year. Both software and hardware.
- Communication Emails will be sent to all users when patching and upgrading is needed
 - Notifications will be sent as early as possible to allow users to plan and prepare.
 - Notifications will include the type of change, the date and time the change will be made, the duration of the change, and if the change requires downtime or an interruption to service.
 - Users will need to communicate this info with their business application teams.
- A Deployment Calendar will be created to track all patches, upgrades and deployments.

Patches and Upgrades (continued)

- Users will need to plan and prepare for patches and upgrades. This could include regression testing, code updates, application upgrades/patches etc...
- Users will need to ensure their code still functions as expected after upgrades and patches:
 - Regression testing
 - Updates to code as needed
 - [MUnit](#) – should be used during the implementation of an interface. This will support automated testing of your interface for events like these.

Unplanned Events

- Unplanned downtime will happen occasionally. The Integration Services team will work to resolve these kinds of events immediately.
- In some cases the Integration Services team will need the support of our users to help resolve issues – if an interface your team owns is causing system wide interruptions or degradation for example.
- All users will need to maintain an updated support contact list for the Integration Services team

Coding and Design Standards

- The Integration Services team maintains a list of [standards and best practices](#) that developers should follow when possible. Includes:
 - Best practices
 - Maven profiles
 - Naming conventions
 - Use of MUnit
- This list is always evolving
- Users are welcome to contribute!

