ISO Release Management

**Release Process**



# Document Control

## Version History

| **Version** | **Date** | **Author** | **Description** |
| --- | --- | --- | --- |
| 0.1 | 02-Mar-2015 | Stacey Acosta | Initial draft of document. |
| 0.2 |  |  |  |
|  |  |  |  |

## Approval/Authorization

| **Name** | **Role** | **Signature** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Document Control 2

Version History 2

Approval/Authorization 2

Terminology 5

1. ISO Release Management 6

2. Development 6

2.1. Core Engine 6

2.2. Plugin 6

2.3. Database 6

2.4. Graphical User Interface (GUI) 6

2.5. Installation 6

3. Confluence Development Handoff 7

4. Confluence Pull to Quality Assurance 7

5. Quality Assurance Testing 7

5.1. Library 7

5.2. Outcomes 7

**List of Tables**

Table 1: REST API Categories and Modules 8

Table 2: CoA Table Permission 10

Table 3: Lookup Table Permission 11

**List of Figures**

Figure 1: User Role Hierarchy 6

Figure 2: Illustration of the ISO Navigation 7

Figure 3: How to execute CoAs within Case 9

Copyright

Copyright © 2015 CSG Cyber Solutions, Inc., and/or its affiliates (“Invotas”). All rights reserved. This software and related material is confidential and proprietary to Invotas and may not be disclosed, copied, modified, or otherwise used except in accordance with the license agreement entered into with Invotas. This information is protected by international copyright laws and any unauthorized use thereof may violate copyright, trademark, and other laws. Any unauthorized use of this software and/or information will automatically terminate your right to use this software and/or information.

Disclaimer Notice

The information contained within this document or application is the property of Invotas, which is confidential and protected by international copyright laws and any unauthorized use of this document or application or its contents may violate copyright, trademark, and other laws. No part of this document or application may be photocopied, reproduced or translated in any form or by any means, or stored in a retrieval system or transmitted electronically or otherwise, without the prior written consent of Invotas.

If you breach any of these terms, your authorization to use this document or application automatically terminates. You may not modify this document or application or its contents in any way or publicly display, perform, or distribute or otherwise use this document or application or its contents for any public or commercial purpose. Any use of this document or application or its contents for any other purpose other than as mutually agreed upon with Invotas is prohibited.

Although every endeavor has been made to ensure that the information contained within this document or application is up to date and accurate, Invotas cannot be held responsible for any inaccuracy or error in the information contained within this document or application. Invotas makes no warranty of any kind with regard to the information and Invotas shall not be liable for any direct, indirect, incidental or consequential damages which may arise in connection with the furnishing, reliance or use of the information contained within this document or application.

Specifications and statements as to performance in this document or application are Invotas estimates, intended for general guidance. Invotas reserves the right to change the information contained within this document or application and any product specification and/or availability dates without notice.

Statements in this document or application are not part of a contract or program product license. Issue of this document or application does not entitle the recipient to access or use of the products described, and such access or use shall be subject to separate contracts or licenses.

Trademark Notice

Invotas Security Orchestrator™ is a trademark of Invotas, and all associated designs and trade names are trademarks of Invotas, and/or its affiliate companies.

CSG Invotas™ and CSG® are trademarks of CSG Systems International, Inc., and all associated designs and trade names are trademarks of CSG Systems International, Inc. and/or affiliate companies.

All third party trademarks, service marks, and/or product names which are referenced in this document are the property of their respective owners, and all rights therein are reserved.

# Terminology

|  |  |
| --- | --- |
| Term | Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. ISO Release Management

This document describes the overall release process as code and documentation moves from origination to a shippable solution.

1. Development
   1. Core Engine

This group develops the core engine code. They will also be responsible for any documentation necessary for installation and initialization of the core engine. Currently this includes scripting for Interactivate.

If Interactivate does not automatically install core engine scripts then this group will also need to provide component installation scripts.

* 1. Plugins

This group develops all plugin code. They will also be responsible for any documentation necessary for installation and initialization of any plugins developed. Some plugins may be for a specific customer or can be for the plugin library.

How plugins are bundled will impact installation of plugins. This group may need to provide installation scripts per plugin or per bundle.

A plugin may have a specific set of instructions for installation and/or configuration. Current discussions include interactive scripts that help with the install process, possible configuration screens, or interactive scripts.

This group will help maintain the index of plugin versions and with which core engine versions each will function.

* 1. Database

This group develops the code for the database and data migration between versions. They will also be responsible for any documentation necessary for installation and initialization of the database.

This group will also need to provide any component installation scripts that the master install script will execute.

* 1. Graphical User Interface (GUI)

This group develops the GUI code for the ISO event management as well as the maintenance application. They will also be responsible for any documentation necessary for installation and initialization of both interfaces.

Installation?

* 1. Installation

This group maintains and updates the installation scripts as needed. They will also be responsible for the content of the Installation Guide.

While this group will not create the scripts to install the individual components, it will be responsible for the master install scripts that execute any component install scripts.

Component management will be here as well. Remove from each section above.

1. Confluence Development Handoff

This is the first handoff but will include the 5 areas of development. Each development group will provide both the code and associated documentation. Each release will have a specific landing zone on Confluence with the corresponding release version number.

All documents should use the Invotas branded template, be properly formatted, and complete in content. (Add Sharepoint link for template – Rhoda will post it to Sharepoint)

List of docs? Or maybe under each section in 2.0

A handoff checklist and/or list of files should accompany the code and documentation. This document will be a living document as the release moves through the Quality Assurance and Release Management procedures.

1. GitHub Pull to Quality Assurance

What will be the build rate: daily, weekly, per sprint?

Will there be patch builds?

How will these be pulled into QA and tested?

Will GitHub replace SVN? If so section 3 content will be put here.

Repositories:

1. GUI,
2. Engine,
3. Plugins (potentially multiples here),
4. Installation,
5. Releasable documentation
6. Confluence/GitHub Pull to Quality Assurance

Lab manager (?) will pull all the code and install scripts from confluence to a clean QA test virtual machine (VM). Verification that all files listed in documentation are present and run the master install script.

Will there be multiple VMs to house the different components (GUI, database, etc)?

At a later date there will be a migration path so testing installs on a clean VM will be in parallel to upgrading the previous version on another VM.

1. Quality Assurance Testing
   1. Library

Any test library that is created will likely be housed on its own virtual machine. This will protect the library as the QA machines are volatile.

Until a library can be built most tests will likely be manual.

* 1. Intervals

Any interval short of an official release will have a slightly different process as the development will still be active. Testing may reveal the need to continue working on something. QA will work with development in deciding what is best for that sprint.

* 1. Outcomes

Hopefully the Security Architects will be able to provide some guidance for expected outcomes and testing. (Rhoda, what doc will this info be in?)

Outcomes will fall into three categories:

1. Passed and accepted
2. Failed but accepted and documented. These issues will be documented as a “Known Issue” in the Release Notes and stories created in JIRA for future follow up.
3. Failed and Rejected and documented. These issues will trigger a rejection of the build. Stories will be returned to developer with associated notes.

Bug Tracking – external system (Bugzilla?), bugs based on version,