

**Prolog New user Request- automation**

**Solution Design Document**

Prepared for Jacobs Engineering

July 13, 2018

**Statement of Confidentiality**

The information contained in this document is confidential to Jacobs. It shall not be disclosed, duplicated, or used for any purpose other than that stated herein, in whole or in part, without the express written consent of Jacobs.

**Document Classification**

|  |  |  |
| --- | --- | --- |
| Public | Internal Use only | Confidential |

|  |  |
| --- | --- |
| X | Do not forward or copy data in part or full without explicit permission of Jacobs |
|  | Data access is limited to \_\_\_\_\_\_\_\_\_\_\_ (URL of list) |
|  | Use Strong authentication/Lock in a Drawer |
|  | Log access in a register |
| X | Retention period is \_\_1\_\_ Year(s) |

**Table of Contents**

[Glossary 4](#_Toc519176683)

[1. Document Control 5](#_Toc519176684)

[1.1 Revision History 5](#_Toc519176685)

[2. Executive Summary 5](#_Toc519176686)

[2.1 Overview 5](#_Toc519176687)

[2.2 Purpose 5](#_Toc519176688)

[2.3 Assumptions 6](#_Toc519176689)

[2.4 Configuration Needs 6](#_Toc519176690)

[2.5 Known issues 6](#_Toc519176691)

[2.6 Deferred Items 6](#_Toc519176692)

[2.7 Reporting Needs 6](#_Toc519176693)

[3. Design Overview of Catalog Item 6](#_Toc519176694)

[3.1 Service Catalog Configuration Information 7](#_Toc519176695)

[3.1.1 Navigational Categorization 7](#_Toc519176696)

[3.1.2 Service Catalog Item Details 7](#_Toc519176697)

[3.1.3 Approval Groups 8](#_Toc519176698)

[3.1.4 Entitlement Groups 8](#_Toc519176699)

[3.1.5 SLA 8](#_Toc519176700)

[3.1.6 Fulfillment Provider(s) 8](#_Toc519176701)

[3.2 Service Catalog Process Flow 9](#_Toc519176702)

[3.3 Service Catalog User Interface Details 9](#_Toc519176703)

[3.4 Implementation Approach 13](#_Toc519176704)

[3.5 User Interface Screen 14](#_Toc519176705)

[4. Automation Overview 15](#_Toc519176706)

[4.1 Soap Message Details. 15](#_Toc519176707)

[4.2 Populating Staging Tables. 16](#_Toc519176708)

[4.3 Schedule Job Execution 17](#_Toc519176709)

[4.4 Functionality/Background Process 17](#_Toc519176710)

[5. SLA References 18](#_Toc519176711)

[6. Future Recommendations 18](#_Toc519176712)

# Glossary

|  |  |
| --- | --- |
| **Word/Abbreviation** | **Description** |
| GUID | Globally Unique Identifier |
| RITM | Requested Item |
|  |  |
|  |  |
|  |  |

# Document Control

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Kubra Iram  Ponnusamy Subramaniam | **06-July-2018** |  | **1.0** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table 1: Revision History**

# Executive Summary

## Overview

Jacobs is looking to automate the User creation in Prolog application using ServiceNow. This Service Offering will enable the users to create new users associated to single or multiple projects. The Catalog Item must be managed in an efficient way by having it configured with appropriate approvals. The implementation of the new Catalog Item will result in with the below benefits.

* Intuitive User Interface.
* Automation of Users Creation.
* Defined approval process.
* Accountability and Ownership
* Quick turnaround.
* Ease of use.
* Controlled Notifications

Jacobs is making use of Prolog Application for their Users Creation. Considering the volume of Users creation by Jacobs, it’s really mandatory to automate their current process as User creation manually leads to time consumption, usage of man power and lack of accountability. The above said challenges has driven Jacobs to lookout for a quicker scalable and robust solution to automate their existing process so that it will be more efficient and will add more value for the organization by having their Users created with proper controls/approvals in place.

## Purpose

The purpose of this document is to identify and capture the design details for “Prolog New User Request - automation” on the Business requirement document.

This document intends to provide a set of Technical details in order to ensure a common understanding of the functionality to be developed. This will serve as the guide for developers.

## Assumptions

1. All the Service catalog Items intended to develop using Maintain Items/Variable Sets/Workflow/Single step checkout process.
2. All the supports groups for Approval / Fulfillments /Entitlement should be available in ServiceNow foundation data.
3. Custom SLA configuration details will be provided by Jacobs.
4. Catalog/Category/Item Images would be provided by Jacobs if required to be displayed against the Item/Category or Catalog.
5. Jacobs will be providing the required Prolog web services along with the credentials.
6. Custom tables will be created to store the Master Data required for this automation.

## Configuration Needs

Based on the requirements, Administrators would make the necessary configurations if requirements having anything to be derived by any configuration form/table (Choice List Table).

## Known issues

N/A at this moment

## Deferred Items

## Reporting Needs

N/A

# Design Overview of Catalog Item

This document is created and extended in Construction phases over the course of the project

* Construction Phase – During the Construction Phase, this document is not expected to change radically; it is mainly updated to reflect changes in any interface definitions.
* Transition / Training Phase – During the Transition/Training Phase, no further additions or modifications are made to this document.

The overall architecture goals of the system is to provide a highly available and scalable ServiceNow Management Catalog offerings for users of the Jacobs, to understand what services are available and to determine if they are potentially eligible for those services.

A key Architectural goal is to leverage industry best practices for designing and developing a scalable, Service offerings with various ITSM fulfilment operations using ServiceNow Platform. To meet this goal, the design of the Service offerings will be based on ServiceNow Best Practices and the industry standard development guidelines for building the Service offerings.

## Service Catalog Configuration Information

The below tables provide the Configuration Information

* + 1. Navigational Categorization

|  |  |
| --- | --- |
| **Field** | **Input Value** |
| Catalog | Click Here to Request a Service from the Catalog |
| Category | Software |
| Item | Prolog New User Request |

* + 1. Service Catalog Item Details

|  |  |  |
| --- | --- | --- |
| **Field** | **Input Value** | **Notes** |
| Definition ID | Defaulted value |  |
| Version | Defaulted value |  |
| Status\* | Defaulted value |  |
|  |  |  |
| Instructions |  |  |
| Company\* | Jacobs |  |
| Locale | Defaulted value |  |
| Item Name | Prolog New User Request |  |
| Description\* | Completion of this form will initiate a User creation.  If an issue is experienced, please contact the Global Service Desk by opening an incident at [http://support.jacobs.com](http://support.jacobs.com/).   If it is urgent phone the Global Service Desk at the appropriate number listed at [http://globalservicedesk.jacobs.com](http://globalservicedesk.jacobs.com/).    **Approval Process** - This requires Functional/Operations approval. If you are requesting an item that falls outside of Jacobs standards, IT may require additional approvals and/or business case and cost justification information prior to proceeding with your request. |  |

* + 1. Approval Groups

|  |  |
| --- | --- |
| **Field** | **Input Value** |
| Group 1 | L3-EAS Supply Management Services: Functional Support |
| Group 2 |  |
| Group 3 |  |
| Group 4 |  |

* + 1. Entitlement Groups

|  |  |
| --- | --- |
| **Field** | **Input Value** |
| Group 1 |  |
| Group 2 |  |
| Group 3 |  |
| Group 4 |  |

* + 1. SLA

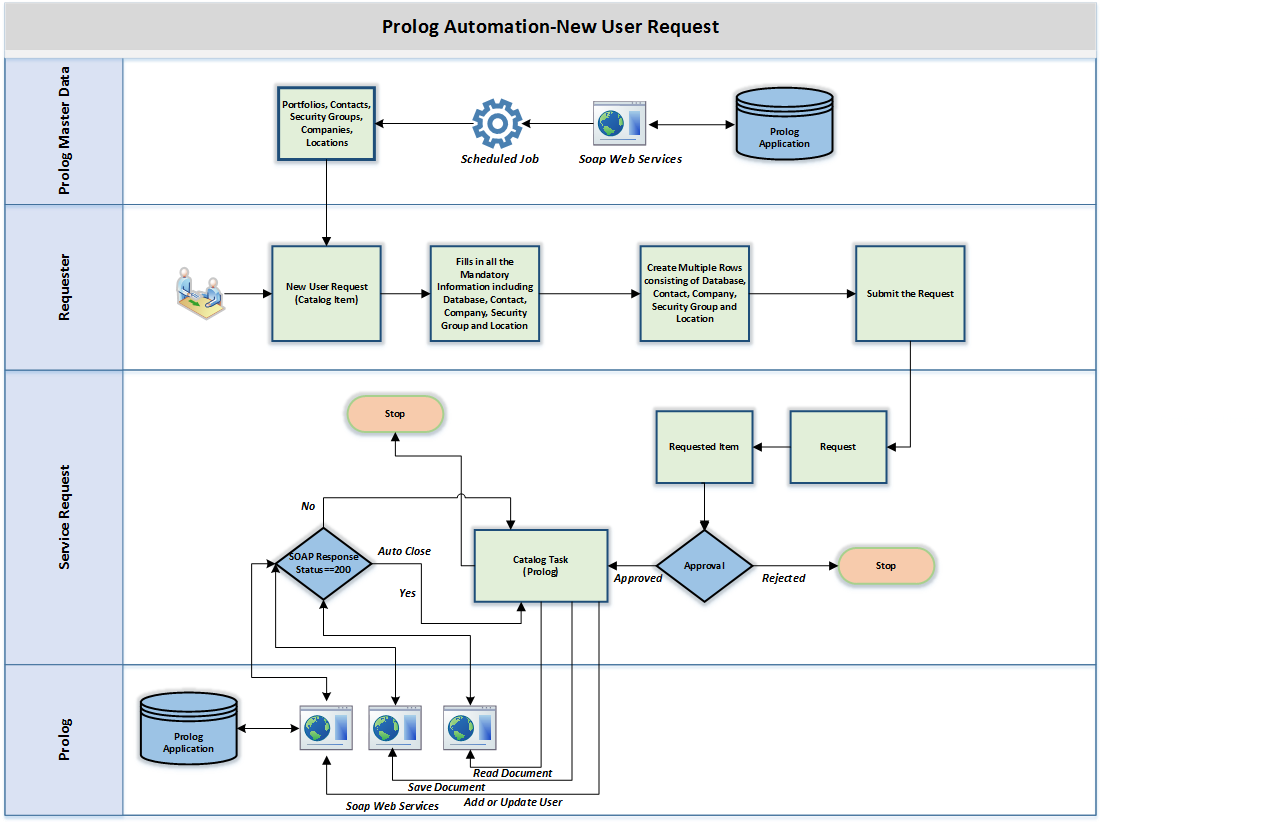
|  |  |
| --- | --- |
| **Field** | **Input Value** |
| Group 1 |  |
| Group 2 |  |

* + 1. Fulfillment Provider(s)

|  |  |
| --- | --- |
| **Field** | **Input Value** |
| Group | L2-EAS Supply Management Services |

## Service Catalog Process Flow

The below process flow depicts how the SR flows from Initiation to completion.



## Service Catalog User Interface Details

The Service Catalog- User Interface is developed using Maintain Items, Variable sets, workflow, UI Policies, Catalog Client Scripts, and Catalog UI Policies or in any combinations of the above, they can be personalized based on the Information provided to the developers.

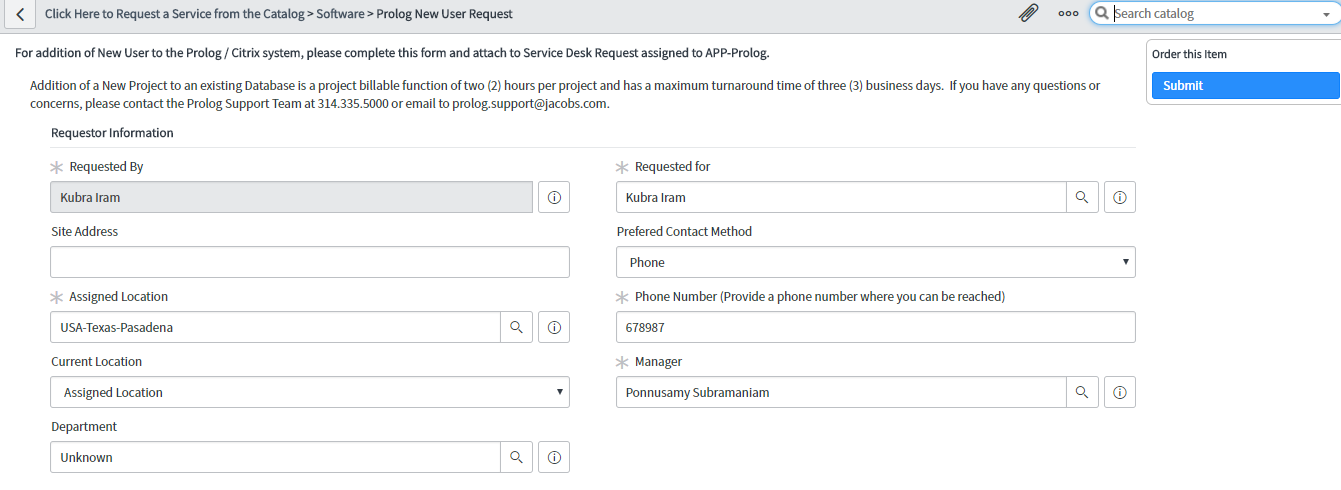
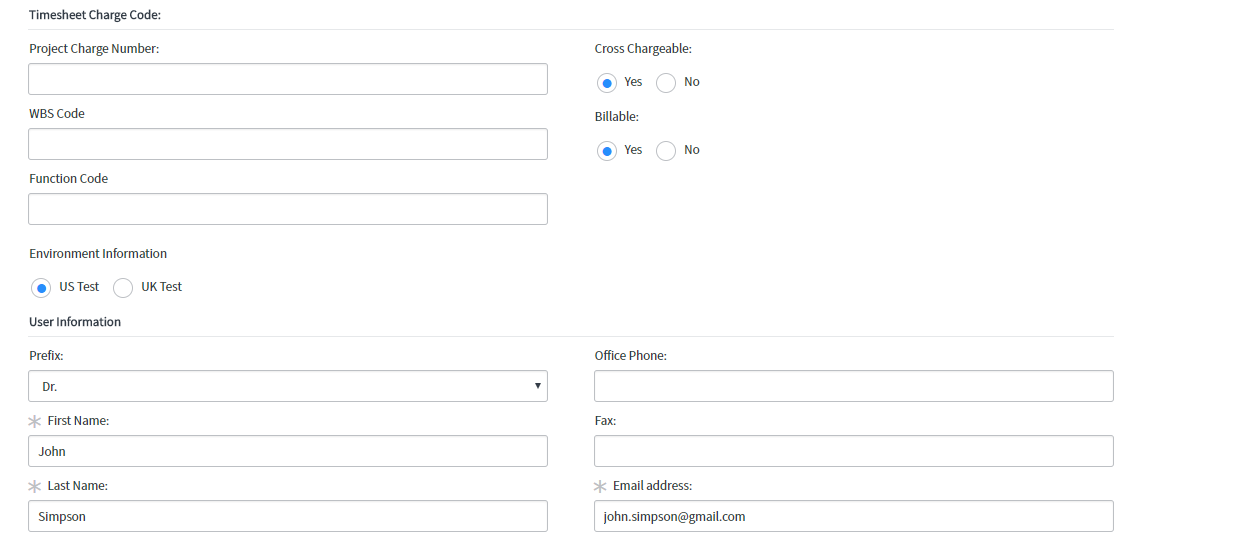
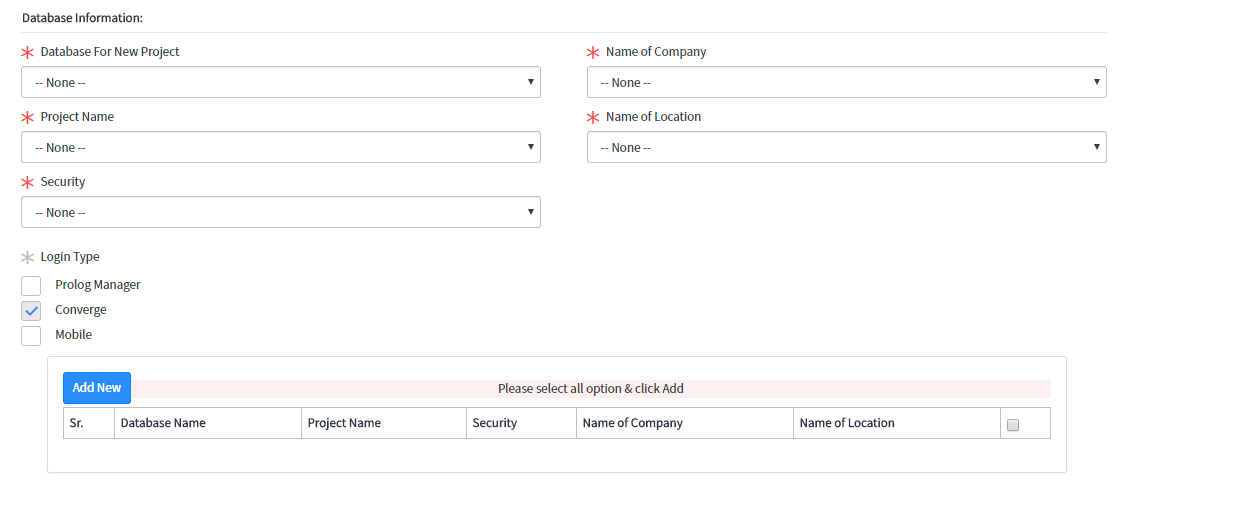
The new Catalog Item contains below fields (questions) shown in the table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | **Question Variables** | **Data Type/Values** | **Description of the question/Input** | **Mandatory Yes/No** | **Reference Table and Question conditions if any** |
| 1 | Requestor Information | Variable Set | Variable set containing details of requestor like Requested By, Requested for, Address, Location, Phone Number, Manager and Department details | Yes |  |
| 2 | Timesheet Charge Code: | Label |  | NO |  |
| 3 | Project Charge Number: | Single Line Text |  | NO | Manual Entry |
| 4 | WBS Code | Single Line Text | Manual Input | NO | Manual Entry |
| 5 | Function Code | Single Line Text | Manual Input | NO | Manual Entry |
| 6 | Cross Chargeable: | Multiple Choice |  | NO | Manual Entry |
| 7 | Billable: | Multiple Choice |  | NO | Manual Entry |
| 8 | Environment Information | Multiple Choice |  | NO | Manual Entry |
| 9 | User Information | Label |  | NO |  |
| 10 | Prefix: | Select Box |  | NO | Choices are as below:   * Dr. * Mr. * Mrs. * Ms. |
| 11 | First Name: | Single Line Text |  | YES | Manual Entry |
| 12 | Last Name: | Single Line Text |  | YES | Manual Entry |
| 13 | Office Phone: | Single Line Text |  | NO | Manual Entry |
| 14 | Fax: | Single Line Text |  | NO | Manual Entry |
| 15 | Email address: | Single Line Text |  | YES | This field is validated for proper format and verified with the staging table to check the redundancy |
| 16 | Database Information: | Label |  | NO |  |
| 17 | Database For New Project | Lookup Select Box |  | YES | u\_ritm0\_626623\_portfolios\_1  (This is a Master data table which is maintained by regular soap calls from the source and dependent on 8) |
| 18 | Project Name | Lookup Select Box |  | YES | Prolog Projects [u\_prolog\_projects] , (This is a Master data table which is maintained by regular soap calls from the source and dependent on 17 and 8) |
| 19 | Security | Lookup Select Box |  | YES | Prolog SecurityGroups [u\_prolog\_securitygroups] (This is a Master data table which is maintained by regular soap calls from the source and dependent on 17 and 8) |
| 20 | Name of Company | Lookup Select Box |  | YES | Prolog Contacts [u\_prolog\_contacts] (This is a Master data table which is maintained by regular soap calls from the source and dependent on 17 and 8) |
| 21 | Name of Location | Lookup Select Box |  | YES | Prolog Contacts [u\_prolog\_contacts] (This is a Master data table which is maintained by regular soap calls from the source and dependent on 17 and 8) |
| 22 | Login Type | Label |  | YES |  |
| 23 | Prolog Manager | CheckBox |  | NO |  |
| 24 | Converge | CheckBox |  | NO | This field is always checked to true and read-only |
| 25 | Mobile | CheckBox |  | NO |  |
| 26 | Record Table | UI Page |  | NO |  |
| 27 | Grid | UI Page |  | NO |  |
| 28 | Site Address | Wide Single Line Text |  | NO |  |
| 29 | Assigned Location | Reference |  | YES | Location [cmn\_location] |
| 30 | Current Location | Select Box |  | NO |  |
| 31 | Alternate Office | Reference |  | YES | Location [cmn\_location] Visible when Current Location=Alternate Location |
| 32 | Department | Reference |  | NO | Location [cmn\_location] |
| 33 | Requested for | Reference |  | YES | User [sys\_user] |
| 34 | Prefered Contact Method | Select Box |  | NO | Default value – ‘Phone’ |
| 35 | Phone Number (Provide a phone number where you can be reached) | Single Line Text |  | YES |  |
| 36 | Manager | Reference |  | YES | User [sys\_user] |
| 37 | Requested By | Reference |  | YES | User [sys\_user] Auto populated |

## Implementation Approach

1. Creation of an Update Set by Name ‘%DATE% TRZ JACOBS New User Request’.
2. Creation of Approval/Entitlement/SLA/Fulfillment groups based on the data specified in the above tables.
3. Check for the existence of a Catalog by Name ‘Click here to Request a Service from the Catalog’ and if not found, create a new one using the ‘Maintain Catalog’ interface.
4. Check for the existence of a Category by Name ‘Software’ and if not found, create a new one using the ‘Maintain Categories’ interface.
5. Create a New Catalog Item with name ‘Prolog New User Request’ with Catalog and Category set to ‘Click here to Request a Service from the Catalog’ and ‘Software’ respectively.
6. Create the variables mentioned in the table 3.3 Service Catalog- User Interface Details.
7. The Entitlements will be configured in the Catalog Item in accordance to the values specified in the above Entitlement table.
8. Create a workflow by name ‘SC - Prolog New User Request’ and map the same in Workflow attribute in Catalog Item table.
9. The Workflow will have activities included for Approval generation and Fulfillment Task creation.
10. Create necessary staging tables and populate the master data using the SOAP webservices provided by Jacobs Prolog team. This is explained in detail in section 4.0.

## User Interface Screen

# Automation Overview

## Soap Message Details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Region** | **End Point** | **Function Name** | **Soap Message Name** |
| 1 | US | <https://convergeustest2.jacobs.com/Prolog/connect/siteinformationservice.asmx> | SiteInformationServiceSoap.GetPortfolio | US\_RITM0 626623-SIS-Get Portfolio |
| 2 | UK | <https://convergeeutest.jacobs.com/Prolog/connect/siteinformationservice.asmx?wsdl> | SiteInformationServiceSoap.GetPortfolios | EU\_RITM0 626623-SIS-Get Portfolio |
| 3 | US | <https://convergeustest.jacobs.com/Prolog/connect/QueryService.asmx?wsdl> | [QueryServiceSoap.](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=687b830adb3ad70460b938ff9d961929&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=38&sysparm_record_list=soap_message%3Dbeca4f86db3ad70460b938ff9d961939%5EORDERBYDESCsys_updated_on) ExecuteDynamicQueryOnPortfolio  Datagroup = 112 | [Prolog\_UserRequest\_GetContact\_US](https://trianzjacobsdev.service-now.com/sys_soap_message.do?sys_id=beca4f86db3ad70460b938ff9d961939&sysparm_record_target=sys_soap_message&sysparm_record_row=5&sysparm_record_rows=50&sysparm_record_list=ORDERBYDESCsys_updated_on) |
| 4 | UK | <https://convergeeutest.jacobs.com/Prolog/connect/QueryService.asmx?wsdl> | [QueryServiceSoap.](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=687b830adb3ad70460b938ff9d961929&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=38&sysparm_record_list=soap_message%3Dbeca4f86db3ad70460b938ff9d961939%5EORDERBYDESCsys_updated_on) ExecuteDynamicQueryOnPortfolio  Datagroup = 112 | Prolog\_UserRequest\_GetContact\_UK |
| 5 | US | <https://convergeustest.jacobs.com/Prolog/connect/QueryService.asmx?wsdl> | [QueryServiceSoap.](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=687b830adb3ad70460b938ff9d961929&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=38&sysparm_record_list=soap_message%3Dbeca4f86db3ad70460b938ff9d961939%5EORDERBYDESCsys_updated_on) ExecuteDynamicQueryOnPortfolio  Datagroup = 39 | [Prolog\_UserRequest\_GetProjectID\_US](https://trianzjacobsdev.service-now.com/sys_soap_message.do?sys_id=643e4390db47534460b938ff9d961966&sysparm_record_target=sys_soap_message&sysparm_record_row=6&sysparm_record_rows=50&sysparm_record_list=ORDERBYDESCsys_updated_on) |
| 6 | UK | <https://convergeeutest.jacobs.com/Prolog/connect/QueryService.asmx?wsdl> | [QueryServiceSoap.](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=687b830adb3ad70460b938ff9d961929&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=38&sysparm_record_list=soap_message%3Dbeca4f86db3ad70460b938ff9d961939%5EORDERBYDESCsys_updated_on) ExecuteDynamicQueryOnPortfolio  Datagroup = 39 | Prolog\_UserRequest\_GetProjectID\_UK |
| 7 | US | <https://convergeustest.jacobs.com/Prolog/connect/QueryService.asmx?wsdl> | [QueryServiceSoap.](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=687b830adb3ad70460b938ff9d961929&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=38&sysparm_record_list=soap_message%3Dbeca4f86db3ad70460b938ff9d961939%5EORDERBYDESCsys_updated_on) ExecuteDynamicQueryOnPortfolio  Datagroup = 950 | [Prolog\_UserRequest\_GetSecurityGroups\_US](https://trianzjacobsdev.service-now.com/sys_soap_message.do?sys_id=19f2a798db87534460b938ff9d9619e4&sysparm_record_target=sys_soap_message&sysparm_record_row=4&sysparm_record_rows=50&sysparm_record_list=ORDERBYDESCsys_updated_on) |
| 8 | UK | <https://convergeeutest.jacobs.com/Prolog/connect/QueryService.asmx?wsdl> | [QueryServiceSoap.](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=687b830adb3ad70460b938ff9d961929&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=38&sysparm_record_list=soap_message%3Dbeca4f86db3ad70460b938ff9d961939%5EORDERBYDESCsys_updated_on) ExecuteDynamicQueryOnPortfolio  Datagroup = 950 | Prolog\_UserRequest\_GetSecurityGroups\_UK |
| 9 | US | https://convergeustest2.jacobs.com/Prolog/connect/DocumentService.asmx?wsdl | [DocumentServiceSoap.ReadDocument](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=c5ed74b7dbfe9b4829253220ad961993&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=26&sysparm_record_list=soap_message%3D1c868423dbfa1b4829253220ad96199a%5EORDERBYDESCsys_updated_on) | Prolog\_UserRequest\_\_ReadSaveDocument\_US |
| 10 | US | https://convergeustest2.jacobs.com/Prolog/connect/DocumentService.asmx?wsdl | [DocumentServiceSoap.SaveDocument](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=24ed74b7dbfe9b4829253220ad96197e&sysparm_record_target=sys_soap_message_function&sysparm_record_row=2&sysparm_record_rows=26&sysparm_record_list=soap_message%3D1c868423dbfa1b4829253220ad96199a%5EORDERBYDESCsys_updated_on) | Prolog\_UserRequest\_\_ReadSaveDocument\_US |
| 11 | US | https://convergeustest2.jacobs.com/Prolog/connect/administrationservice.asmx?wsdl | [AdministrationServiceSoap.AddOrUpdateUser](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=ecfe3d77db7adb4829253220ad9619ca&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=64&sysparm_record_list=soap_message%3Dca09c823dbfa1b4829253220ad9619c3%5EORDERBYDESCsys_updated_on) | Prolog\_UserRequest\_AddUpdateUser\_US |
| 12 | UK | https://convergeeutest2.jacobs.com/Prolog/connect/DocumentService.asmx?wsdl | [DocumentServiceSoap.ReadDocument](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=c5ed74b7dbfe9b4829253220ad961993&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=26&sysparm_record_list=soap_message%3D1c868423dbfa1b4829253220ad96199a%5EORDERBYDESCsys_updated_on) | Prolog\_UserRequest\_\_ReadSaveDocument\_UK |
| 13 | UK | https://convergeeutest2.jacobs.com/Prolog/connect/DocumentService.asmx?wsdl | [DocumentServiceSoap.SaveDocument](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=24ed74b7dbfe9b4829253220ad96197e&sysparm_record_target=sys_soap_message_function&sysparm_record_row=2&sysparm_record_rows=26&sysparm_record_list=soap_message%3D1c868423dbfa1b4829253220ad96199a%5EORDERBYDESCsys_updated_on) | Prolog\_UserRequest\_\_ReadSaveDocument\_UK |
| 14 | UK | https://convergeeutest2.jacobs.com/Prolog/connect/administrationservice.asmx?wsdl | [AdministrationServiceSoap.AddOrUpdateUser](https://trianzjacobsdev.service-now.com/sys_soap_message_function.do?sys_id=ecfe3d77db7adb4829253220ad9619ca&sysparm_record_target=sys_soap_message_function&sysparm_record_row=1&sysparm_record_rows=64&sysparm_record_list=soap_message%3Dca09c823dbfa1b4829253220ad9619c3%5EORDERBYDESCsys_updated_on) | Prolog\_UserRequest\_AddUpdateUser\_UK |

## Populating Staging Tables.

* There are 4 staging table’s

1) Portfolio Table (RITM0 626623-PortFolios-1)

2) Project IDs Table (u\_prolog\_projects)

3) Contacts Table (u\_prolog\_contacts)

4) Security Groups Table (u\_prolog\_securitygroups).

* The data from the prolog system is collected in the above staging tables through schedule jobs that are executed every 30 minutes.
* These schedule jobs contains soap message scripts that pulls the data from the prolog system and populates it in the staging table.
* The overview of the end point and the soap function used in the Soap Messages US and UK Regions are mentioned above. The complete details are provided in a separate word document detailing sample requests. (Attached below)



* The portfolio name and first entry of the project in the portfolio is extracted and populated in the portfolio staging table.
* To extract the Project IDs, the portfolios from the staging table is passed in the Project ID request to extract corresponding template entries.
* The same is done to populate Contact and Security Groups staging tables.

## Schedule Job Execution

* There are 4 schedule jobs created per region, to populate the data into staging tables.
* The schedule job related to portfolio staging table runs every 30 minutes.
* The portfolio schedule job contains a script which triggers the Project ID schedule job. This is triggered once the data in the portfolio staging table is populated.
* The Project ID schedule job in turn triggers the Contacts schedule job the same way as portfolio is triggered.
* In the similar way the Security Groups staging table gets its data populated.
* Similar structure is followed for EU Schedule job runs.

## Functionality/Background Process

* A scheduled job is scheduled to run once in 30 minutes to load Database (Portfolio), Projects, Contacts and Security Groups from Prolog into ServiceNow using SOAP Endpoints.
* "A Requestor creates a request using the Prolog New User Catalog Item filling in all the Mandatory values. The Requestor should be allowed to create multiple rows containing Database Name, Project Name, Security, Company, and Location values and the same will get added to a Grid at the bottom.
* The combined column values in each row should be unique.
* The Database Name field is dependent on the Environment field.
* The Project Name, Security, Company, and Location field values are dependent on the Database Name field.
* The Requestor should be having an option to remove any rows added.
* The System must validate for atleast one row in the grid before submitting the request."
* A Requested Item should be created part of the Service Request.
* A group approval should be generated and it will be sent to all members of IT-L3-EAS Supply Management Services: Functional Support group.
* Once RITM is approved by anyone member of the approval group, then a corresponding Catalog Task will be created.
* If RITM is Rejected, then the state of the corresponding RITM should be set to Closed Complete with no Catalog Task created. The Request ends here.
* The Catalog Task should trigger SOAP calls to post the information submitted by the user to Prolog. Below SOAP calls will be triggered in sequence.
  + **Read Document ->** This soap call is triggered with Database name, Project Name and Company GUID (This information is captured in Contacts staging table) as parameters and in turn it will return the <Document\_data> in response.
  + **Save Document ->** This soap call is triggered with Database name, and the <Document\_data> details from “Read Document” response, and new User Information along with Project detail information in the Security Grid.
  + **AddOrUpdateUser ->** This soap call is triggered with Database name, new User Information and Password, Project and Security Group details.
* The above calls are triggered in iteration based on no. of line items in security grid.
* The Catalog Task should be auto closed if the return status code is 200. This in turn will auto close the RITM and Request as well.
* In case of any errors part of SOAP calls, the same will be captured part of the Catalog Task. The Catalog Task should be updated with state set to 'Open' and assigned to IT-L2-Remote Service Delivery group.
* The Task should be picked up by a member from the assignment group and manually fulfilled.
* No specific Notifications are needed for this particular Catalog Item. The ones available configured for Catalog Items applies to this as well.

# SLA References

|  |  |
| --- | --- |
| SLA Duration | NA |
| SLA Schedule | NA |
| SLA Time Zone | NA |
| ITEM Level SLA Start Condition | NA |
| ITEM Level SLA Pause Condition | NA |
| ITEM Level SLA Stop Condition | NA |
| SLA % time 0.5 | NA |
| SLA % time 25%(75%) | NA |

# Future Recommendations

N/A