Pride – Replenishment List/Smart Stockroom



Updated: 6/7/2019

This document contains proprietary information and should be shared only with appropriate audiences.

**Table of Contents**

[System Description: 5](#_Toc10800813)

[General Description: 5](#_Toc10800814)

[Application Owner: 5](#_Toc10800815)

[Business Partners: 5](#_Toc10800816)

[Service Manager Support Queues: 5](#_Toc10800817)

[Primary Hours of Operation: 5](#_Toc10800818)

[Volumetric Data (Avg. Incidents, Requests, Changes per month): 5](#_Toc10800819)

[Security: 6](#_Toc10800820)

[*Keystone* 6](#_Toc10800821)

[*MyId* 6](#_Toc10800822)

[Risks: 6](#_Toc10800823)

[Build and Deploy: 6](#_Toc10800824)

[Jenkins: 6](#_Toc10800825)

[github: 6](#_Toc10800826)

[Sonar: 7](#_Toc10800827)

[Installed Environments: 7](#_Toc10800828)

[Production: 7](#_Toc10800829)

[Disaster Recovery: 8](#_Toc10800830)

[Stage: 9](#_Toc10800831)

[Latest: 10](#_Toc10800832)

[System Architecture: 11](#_Toc10800833)

[Pride Component Overview: 11](#_Toc10800834)

[Interfaces: 11](#_Toc10800835)

[pride-ui: 12](#_Toc10800836)

[pride-svc: 12](#_Toc10800837)

[pride-batch: 13](#_Toc10800838)

[UC4 Batches: 13](#_Toc10800839)

[pride-db: 14](#_Toc10800840)

[MyId: 14](#_Toc10800841)

[How To: 14](#_Toc10800842)

[How to validate application health 14](#_Toc10800843)

[Verify Servers: 14](#_Toc10800844)

[Verify pride-svc: 15](#_Toc10800845)

[Verify pride-ui: 16](#_Toc10800846)

[Verify pride-batch: 16](#_Toc10800847)

[How to shut down and restart the Production Pride Applications 16](#_Toc10800848)

[How to change the DB configurations for pride-svc 17](#_Toc10800849)

[How to change the Matra connection information 17](#_Toc10800850)

[How to change the Appetize connection information 17](#_Toc10800851)

[~~How to generate a new Scandit License Key and Deploy (No longer using Scandit but left it the documentation in case we decide to use them in the future).~~ 17](#_Toc10800852)

[How to generate a new DataSymbol License Key and Deploy 17](#_Toc10800853)

[How to update a store to move to appetize 18](#_Toc10800854)

[How re-enroll a mobile device to an AirWatch profile 19](#_Toc10800855)

[Troubleshooting: 20](#_Toc10800856)

[Store selector not showing 20](#_Toc10800857)

[Replenishment List is showing a broken image for the item 21](#_Toc10800858)

[Replenishment List is not showing any items or a particular item 21](#_Toc10800859)

[I am unable to putaway an item 22](#_Toc10800860)

[I am not able to find an item in the item inquiry 22](#_Toc10800861)

[I am not able to see locations in the item inquiry 22](#_Toc10800862)

[Scanning is not working for some devices 22](#_Toc10800863)

[Scanning is not working for all devices 22](#_Toc10800864)

[Disaster Recovery: 22](#_Toc10800865)

[Appendix: 23](#_Toc10800866)

[Data Dictionary Pride DB Tables 23](#_Toc10800867)

[STORE\_ITEM\_BIN\_XREF 23](#_Toc10800868)

[IMPORTANT COLUMNS 23](#_Toc10800869)

[STORE\_ITEM\_BIN\_TRANS\_LOG 23](#_Toc10800870)

[IMPORTANT COLUMNS 23](#_Toc10800871)

[BIN 23](#_Toc10800872)

[IMPORTANT COLUMNS 23](#_Toc10800873)

[BIN\_TRANS\_LOG 23](#_Toc10800874)

[IMPORTANT COLUMNS 23](#_Toc10800875)

[STORE\_ITEM 23](#_Toc10800876)

[IMPORTANT COLUMNS 24](#_Toc10800877)

[STORE 24](#_Toc10800878)

[IMPORTANT COLUMNS 24](#_Toc10800879)

[STORE\_TRANS\_LOG 25](#_Toc10800880)

[IMPORTANT COLUMNS 25](#_Toc10800881)

[ITEM 25](#_Toc10800882)

[IMPORTANT COLUMNS 25](#_Toc10800883)

[ITEM\_CODE 25](#_Toc10800884)

[IMPORTANT COLUMNS 25](#_Toc10800885)

[STORE\_ITEM\_THRESHOLD 25](#_Toc10800886)

[IMPORTANT COLUMNS 25](#_Toc10800887)

[STORE\_ITEM\_THRESHOLD\_TRANS\_LOG 26](#_Toc10800888)

[IMPORTANT COLUMNS 26](#_Toc10800889)

[DEPARTMENT 26](#_Toc10800890)

[IMPORTANT COLUMNS 26](#_Toc10800891)

[DEPARTMENT\_TRANS\_LOG 27](#_Toc10800892)

[IMPORTANT COLUMNS 27](#_Toc10800893)

[PICKLIST 27](#_Toc10800894)

[IMPORTANT COLUMNS 27](#_Toc10800895)

[PICKLIST\_TRANS\_LOG 28](#_Toc10800896)

[IMPORTANT COLUMNS 28](#_Toc10800897)

[ITEM\_DELETES 28](#_Toc10800898)

[POS\_SERVER 28](#_Toc10800899)

[IMPORTANT COLUMNS 28](#_Toc10800900)

# System Description:

## General Description:

This system controls the item replenishment from stockroom to floor as well as item stockroom locations. Pride interfaces with SIM Database and the Point of Sale services (Matra and Appetize).

## Application Owner:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Department** | **Phone** | **Wireless** | **Email** |
| Melanie Roehl | Warehouse Support | 407-828-1072 | 321-228-5571 | Melanie.Roehl@Disney.com |

## Business Partners:

Heart of House (HOH) leaders as well as Store Cast members throughout Walt Disney Parks and Resorts at Walt Disney World and Disneyland (Stockers, Pullers and Store Leaders)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Department** | **Phone** | **Wireless** | **Email** |
| Julie Hotchkiss | Merchandise Integration | 407-397-6001 | 321-436-7323 | [Julie.hotchkiss@disney.com](mailto:Julie.hotchkiss@disney.com) |
| Adrienne Cappuccio | Merchandise Integration | 407-397-6687 | 714-493-6136 | [Adrienne.cappuccio@disney.com](mailto:Adrienne.cappuccio@disney.com) |
| Jennifer Gagon | Retail Operations | 714-781-1656 | 714-448-5816 | [Jennifer.gagon@disney.com](mailto:Jennifer.gagon@disney.com) |

## Service Manager Support Queues:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Queue name** | **Handles** | **Phone** | **Alternate Phone** | **Email** |
| app-global-pride | Primary Application Support | See Oncall | See Oncall | See Oncall |
| ops-global-cicd | CICD Support | See Oncall | See Oncall | See Oncall |
| mls-global-wdprse | Middleware/Loadbalancer | See Oncall | See Oncall | See Oncall |
| dba-global-wdprora | Database DBA Support | See Oncall | See Oncall | See Oncall |
| svr-global- tprlin | Application SA Support | See Oncall | See Oncall | See Oncall |
| app-global-wdprpos | Point of Sale (Matra) | See Oncall | See Oncall | See Oncall |
| app-global-wdpr-pos-appetize | Point of Sale (Appetize) | See Oncall | See Oncall | See Oncall |
| app-global-oretail | SIMBA (SIM) | See Oncall | See Oncall | * See Oncall |
| euc-flwdw-dcomios | Datacom (Device Support) | See Oncall | See Oncall | * See Oncall |
| euc-global-WDPRAirwatch | AirWatch (Device Profile Management) | See Oncall | See Oncall | * See Oncall |

## Primary Hours of Operation:

24 Hours – Supports 24 hour work in stores at Walt Disney World and Disneyland

#### Scheduled Maintenance Hours:

This application resides on SIMBA hardware and follows the outage and maintenance windows as SIMBA

## Volumetric Data (Avg. Incidents, Requests, Changes per month):

Incidents < 1 per month

Changes < 1 per month

RM's < 1 per month

## Security:

Direct server access and permissions are managed by the Pride Support team and subject to approval and regular review by the application owner.

User IDs and application access are managed in Keystone by the super users listed in the “Business Partners” section above.

#### Keystone:

**Application Name:** TWDC.WDPRMerchandise.Pride

**Application Id:** a05aec2e-3f09-451a-ac67-84f98a2113e5

**Production Login:** <https://ui.keystone.disney.com/>

**Stage Login:** <https://ui.keystone-stg.disney.com/>

#### MyId:

**Application Name:** TWDC.WDPRMerchandise.Pride

##### MyId Environments:

MyId is setup as OIDC with a callback including a bearer token to the UI to validate the user credentials and any elevated privileges they may have from Keystone.

**Latest Callback URL:** <https://pride-latest.wdpr.disney.com:443/pride/>?

**Stage Callback URL:** <https://pride-stage.wdpr.disney.com:443/pride/>?

**Prod Callback URL:** <https://pride.wdpr.disney.com:443/pride/>?

**Disaster Recovery Callback URL:** <https://pride-dr.wdpr.disney.com:443/pride/>?

## Risks:

This 24/7 availability of this application is an integral part of the operation of the Walt Disney Parks and Resorts store locations on the East and West coast (WDW and DLR). Down time impacts labor costs and extended periods of down time can impact availability of merchandise on store floor throughout the Walt Disney Parks and Resorts.

## Build and Deploy:

This section provides information on how to build and deploy the application to multiple environments.

### Jenkins:

Jenkins will be used to build, deploy and promote artifacts thru the CICD pipeline for the pride-ui and pride-svc. They are setup for continuous integration and deployment along with zero downtime deploys using standard Disney CICD pipelines.

**pride-ui:** <https://dlp.cicd.wdprapps.disney.com/job/pride-ui/>

**pride-svc:** <https://dlp.cicd.wdprapps.disney.com/job/pride-svc/>

**pride-batch:** There is no CICD pipeline for the batch. This is manually deployed and is called by UC4. The war is deployed manually to: **/opt/apps/WDPRApps/MW\_HOME\_PRIDE/batch.** The log cleanup script (PrideBatchPurgeLogs.sh) needs to be manually copied to **/opt/apps/WDPRApps/MW\_HOME\_PRIDE/scripts.**

### github:

github is used as our source control tool to store application code for the ui, service, batch and database objects.

**pride-ui:** <https://github.disney.com/WDPR-SIMBA/pride-ui>

**pride-svc:** <https://github.disney.com/WDPR-SIMBA/pride-svc>

**pride-batch:** <https://github.disney.com/WDPR-SIMBA/pride-batch>

**pride-db:** <https://github.disney.com/WDPR-SIMBA/pride-db>

### Sonar:

Sonar is the static code analysis tool used to calculate code coverage and code quality for pride-ui and pride-svc.

**pride-ui:** <http://latest-sonar-server.cicd.wdprapps.disney.com:8080/dashboard?id=pride-ui>

**pride-svc:** <https://sonar.cicd.wdprapps.disney.com/dashboard?id=com.wdpr.pride%3Apride-svc>

**pride-batch:** <https://sonar.cicd.wdprapps.disney.com/dashboard?id=com.wdpr.pride%3Apride-batch>

## Installed Environments:

### Production:

**The production application servers are connected to a load balancer which passes traffic to all active servers.**

**pride-ui Load Balancer:** <https://pride.wdpr.disney.com/pride>

**pride-svc Load Balancer:** <https://pride-svc.wdpr.disney.com:8443/pride-svc>

**Load Balancer Dashboard:** <https://pride-svc.wdpr.disney.com:8443//dashboard>

**pride-svc Direct:**

**tomcatA2:**

<https://fldcvpsla17362.wdw.disney.com:8042/pride-svc>

<https://fldcvpsla17363.wdw.disney.com:8042/pride-svc>

**tomcatB2:**

<https://fldcvpsla17362.wdw.disney.com:9042/pride-svc>

<https://fldcvpsla17363.wdw.disney.com:9042/pride-svc>

**UC4: Automic\_v11\_PROD**

**Client: 2400**

**Department: WDW**

|  |  |  |
| --- | --- | --- |
| **Application** | **Server** | **Ports** |
| **pride-ui** | **fldcvpsla1970[3-4]** | **443, 8081** |
| **pride-svc** | **fldcvpsla1736[2-3]** | **8443, 8042, 9042, 8082, 9082** |
| **pride-batch** | **fldcvpsla1736[2-3]** | **8084, 8085, 8086, 8087** |

**Database Schemas: WD\_PICKLIST and SIM  
Database Server:** sbasimpsvc.wdw.disney.com

**TNS:** SIM16P=

(DESCRIPTION=

(ADDRESS=

(PROTOCOL=TCP)

(HOST=fldcppclo11076-scan.wdw.disney.com)

(PORT=1541)

)

(CONNECT\_DATA=

(SERVER=dedicated)

(SERVICE\_NAME=sbasimpsvc.wdw.disney.com)

)

)

**Splunk Application Logs:** <https://wdprt-splunk-prod.wdw.disney.com/en-US/>

**Index=wdpr\_simba**

**sourcetype=ra\_json**

**Splunk Dashboards:**

**Main Overview:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_dashboard>

**Response code Distribution:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_response_code_distribution_dashboard>

**Response time:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_response_time_dashboard>

**Call Volume:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_call_volume_dashboard>

### Disaster Recovery:

**pride-ui Load Balancer:** <https://pride-dr.wdpr.disney.com/pride>

**pride-svc Load Balancer:** <https://pride-svc-dr.wdpr.disney.com:8443/pride-svc>

**Load Balancer Dashboard:** <https://pride-svc-dr.wdpr.disney.com:8443//dashboard>

**pride-svc Direct:**

**tomcatA2:**

<https://nckmvpsla18444.wdw.disney.com:8042/pride-svc>

<https://nckmvpsla18444.wdw.disney.com:8042/pride-svc>

**tomcatB2:**

<https://nckmvpsla18444.wdw.disney.com:9042/pride-svc>

<https://nckmvpsla18444.wdw.disney.com:9042/pride-svc>

**UC4: TODO ADD INFO HERE**

|  |  |  |
| --- | --- | --- |
| **Application** | **Server** | **Ports** |
| **pride-ui** | **nckmvpsla18445** | **443, 8081** |
| **pride-svc** | **nckmvpsla18444** | **8443, 8042, 9042, 8082, 9082** |
| **pride-batch** | **nckmvpsla18444** | **8084, 8085, 8086, 8087** |

**Database Schemas: WD\_PICKLIST and SIM**

**Database Server: TODO ADD INFO HERE**

**DB Name:**  **TODO ADD INFO HERE**

**Splunk Application Logs:** <https://wdprt-splunk-prod.wdw.disney.com/en-US/>

**Index=wdpr\_simba**

**sourcetype=ra\_json**

**Splunk Dashboards:**

**Main Overview:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_dashboard>

**Response code Distribution:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_response_code_distribution_dashboard>

**Response time:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_response_time_dashboard>

**Call Volume:**

<https://wdprt-splunk-prod.wdw.disney.com/en-US/app/search/pride_call_volume_dashboard>

### Stage:

**pride-ui Load Balancer:** <https://pride-stage.wdpr.disney.com/pride>

**pride-svc Load Balancer:** <https://pride-svc-stage.wdpr.disney.com:8443/pride-svc>

**Load Balancer Dashboard:** <https://pride-svc-stage.wdpr.disney.com:8443//dashboard>

**pride-svc Direct:**

**tomcatA2:**

<https://fldcvfsla16937.wdw.disney.com:8042/pride-svc>

<https://fldcvpsla16939.wdw.disney.com:8042/pride-svc>

**tomcatB2:**

<https://fldcvfsla16937.wdw.disney.com:9042/pride-svc>

<https://fldcvpsla16939.wdw.disney.com:9042/pride-svc>

**UC4: Automic\_v11\_NONPROD**

**Client: 2200**

**Department: WDW**

|  |  |  |
| --- | --- | --- |
| **Application** | **Server** | **Ports** |
| **pride-ui** | **fldcvfsla1970[1-2]** | **443, 8081** |
| **pride-svc** | **fldcvfsla1693[7,9]** | **8443, 8042, 9042, 8082, 9082** |
| **pride-batch** | **fldcvfsla1693[7,9]** | **8084, 8085, 8086, 8087** |

**Database Schemas: WD\_PICKLIST and SIM  
Database Server: sbasimisvc.wdw.disney.com**

**TNS: SIM16STAGE= (DESCRIPTION=**

**(ADDRESS=**

**(PROTOCOL=TCP)**

**(HOST=fldcpsclo11074-scan.wdw.disney.com)**

**(PORT=1541)**

**)**

**(CONNECT\_DATA=**

**(SERVER=dedicated)**

**(SERVICE\_NAME=sbasimisvc.wdw.disney.com)**

**)**

**)**

**Splunk Application Logs:** <https://wdprt-splunk-qa.wdw.disney.com/en-US/>

**Index=wdpr\_simba**

**sourcetype=ra\_json**

**Splunk Dashboards:**

**Main Overview:**

<https://wdprt-splunk-qa.wdw.disney.com/en-US/app/search/pride_dashboard>

**Response code Distribution:**

<https://wdprt-splunk-qa.wdw.disney.com/en-US/app/search/pride_response_code_distribution_dashboard>

**Response time:**

<https://wdprt-splunk-qa.wdw.disney.com/en-US/app/search/pride_response_time_dashboard>

**Call Volume:**

<https://wdprt-splunk-qa.wdw.disney.com/en-US/app/search/pride_call_volume_dashboard>

### Latest:

**pride-ui Load Balancer:** <https://pride-latest.wdpr.disney.com/pride>

**pride-svc Load Balancer:** <https://pride-svc-latest.wdpr.disney.com:8443/pride-svc>

**Load Balancer Dashboard:** <https://pride-svc-latest.wdpr.disney.com:8443//dashboard>

**pride-svc Direct:**

**tomcatA2:**

<https://fldcvdsla17356.wdw.disney.com:8042/pride-svc>

**tomcatB2:**

<https://fldcvdsla17356.wdw.disney.com:9042/pride-svc>

**UC4: Automic\_v11\_NONPROD**

**Client: 2000**

**Department: WDW**

|  |  |  |
| --- | --- | --- |
| **Application** | **Server** | **Ports** |
| **pride-ui** | **fldcvdsla17357** | **443, 8081** |
| **pride-svc** | **fldcvdsla17356** | **8443, 8042, 9042, 8082, 9082** |
| **pride-batch** | **fldcvdsla17356** | **8084, 8085, 8086, 8087** |

**Database Schemas: WD\_PICKLIST and SIM  
Database Server: sbasimdsvc.wdw.disney.com**

**TNS: SIM16DEV=**

**(DESCRIPTION=**

**(ADDRESS=**

**(PROTOCOL=TCP)**

**(HOST=fldcpsclo11077-scan.wdw.disney.com)**

**(PORT=1541)**

**)**

**(CONNECT\_DATA=**

**(SERVER=dedicated)**

**(SERVICE\_NAME=sbasimdsvc.wdw.disney.com)**

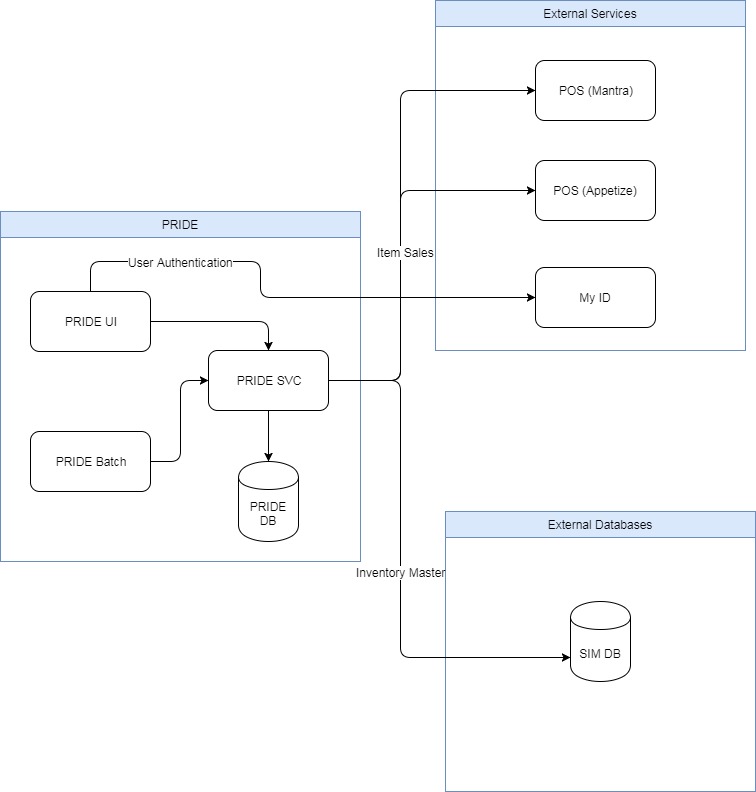
**)**

**)**

**Splunk Application Logs:** N/A

**File Logs: /var/opt/apps/WDPRApps/logs/pride-svc\_tomcat[A/B]2.log**

## System Architecture:



## Pride Component Overview:

### Interfaces:

#### POS

The pride application interfaces with the POS to retrieve the sales data for each store. Each store in the STORE table has a LEGACY\_POS flag. If the flag is true (1) this indicates that we will connect to the legacy system (Matra) to attempt to retrieve store item movement (item sales). If the flag is false (0) this indicates that the store has migrated to the new system (Appetize) to attempt to retrieve the store item movement.

##### Matra:

Matra connections are configured in the POS\_SERVER table. Each store has a POS\_SERVER\_ID to link it back to one of the Matra servers. The POS\_SERVER table will contain the connection information for the store to retrieve the item movement from Matra when the STORE table is setup with the LEGACY\_POS flag set to true (1).

##### Appetize:

Appetize connections are configured in the STORE table. If the store is setup as LEGACY\_POS set to false (0) then the store will connect to the Appetize service configured in the environment property file. Because Appetize does not sum the item movement we pull all orders for the store and sum all item sales for all orders to calculate total item sales for the store.

### pride-ui:

pride-ui is an angular application that runs on nodejs. It connects to MyId to validate the user has a valid Disney login and will pull elevated privileges at this time. The basic project structure for the client contains 3 folders: core, feature, layout as well as a static folder which contains the environment specific configurations. Below is a high level summary for each.

1. Core: contains the code for the core components implementation such as Scandit, account menu, toast messages, etc
2. Feature: contains the code for specific application features such as bin inquiry, item inquiry, picklist, etc.
3. Layout: contains the code for the application layout/look and feel such as images and styles
4. Config: This folder contains a file for each specific environment to hold environment specific configurations such as the pride service, MyId, and Scandit key.

**Development Environment Setup:**

<https://github.disney.com/WDPR-SIMBA/pride-ui>

### pride-svc:

pride-svc spring-boot application that provides REST services to the pride-ui and the pride-batch to connect into any databases or call any external services. This application is the core business logic application. The basic project structure is as follows:

* 1. Controller

🡪 2. Helper

🡪 3. Facade

🡪4. DAO

1. Controller: The entry point into the application. They are @RestController based on spring. They receive the input, validate the input and call other classes to perform business logic.
2. Helper: The base class to contain the majority of the business logic for complex operations. They receive input from the controller can call facades (external services) or DAO’s to retrieve and update data.
3. Facade: The abstraction to connect into other external services. This should contain the code to connect into other services and retrieve the data needed to process the request.
4. DAO: The abstraction to connect into the database tables. This should contain the database queries and the code to dynamically build queries based on input from the service.

Configurations for the service are located in src/main/resources

1. Configurations that are not environment specific are within the properties folder
2. Configurations that are environment specific are with the Config folder and a sub folder for each environment. i.e. Production specific configurations will reside in the PRD subfolder under the Config folder.

**Development Environment Setup:**

<https://github.disney.com/WDPR-SIMBA/pride-svc>

**Service Catalog**

**Documentation for all REST services available and how to call them can be found at the following location.**

<https://github.disney.com/WDPR-SIMBA/pride-svc/tree/develop/Documents>

### pride-batch:

pride-batch is a spring-batch application setup to pull information from SIM and POS to update our view of the inventory data and item movement respectively. The batches are configured and run thru UC4. Mainly the process flow consists of a Reader object to pull and loop through a specific dataset and a Writer object to persist that data. Since the project is structured around REST services interacting with the database, custom Readers have been developed. For batches that don’t require parameters, a Tasklet object is used to call that service.

**Development Environment Setup:**

https://github.disney.com/WDPR-SIMBA/pride-batch

**Service Catalog**

**Documentation for all REST services available and how to call them can be found at the following location.**

<https://github.disney.com/WDPR-SIMBA/pride-batch/tree/develop/Documents>

### UC4 Batches:

**JOBP: Parent jobs that call multiple JOBS**

1. **JOBP.SIMBA\_PRIDE.NIGHTLY\_REFRESH (runs every morning to refresh from SIMBA)**
   1. **JOBS.SIMBA\_PRIDE.DEPARTMENT\_LOAD**
   2. **JOBS.SIMBA\_PRIDE.ITEM\_LOAD**
   3. **JOBS.SIMBA.PRIDE.ITEM\_CODE\_LOAD**
   4. **JOBS.SIMBA\_PRIDE.STORE\_LOAD**
   5. **JOBS.SIMBA\_PRIDE.STORE\_ITEM\_LOAD**

**JOBS: Individual job called by a JOBP or runs independent**

**Called from Parent**

**JOBS.SIMBA\_PRIDE.DEPARTMENT\_LOAD (Loads all Simba departments to Pride)**

**JOBS.SIMBA\_PRIDE.ITEM\_LOAD (Loads all Simba items to Pride by department)**

**JOBS.SIMBA.PRIDE.ITEM\_CODE\_LOAD (Loads all item codes from Simba to Pride)**

**JOBS.SIMBA\_PRIDE.STORE\_LOAD (Loads all stores from Simba to Pride)**

**JOBS.SIMBA\_PRIDE.STORE\_ITEM\_LOAD (Loads all store items from Simba to Pride)**

**Independent**

JOBS.SIMBA\_PRIDE.SERVER\_LOG\_PURGE (called 1x per week to clean old logs)

**JOBS.SIMBA\_PRIDE.BATCH\_LOG\_PURGE\_A (called 1x per week to clean old logs)**

**JOBS.SIMBA\_PRIDE.BATCH\_LOG\_PURGE\_B (called 1x per week to clean old logs)**

**JOBS.SIMBA\_PRIDE.PICKLIST\_LOAD (called every 5 minutes to pull sales from POS)**

**JOBS.SIMBA\_PRIDE.SERVICE\_LOG\_PURGE (called 1x per week to clean old logs)**

**JOBS.SIMBA\_PRIDE.STORE\_DAILY\_RESET (called every hr to reset stores for that hr)**

**JOBS.SIMBA\_PRIDE.ASN\_LOAD (called nightly to pull asn details from SIM to display in Pride)**

### pride-db:

The pride-db is a new schema added to the existing SIM database. This new schema contains application specific information such has bins, replenishment items, store details, item details, etc. The table scripts live in github.

<https://github.disney.com/WDPR-SIMBA/pride-db>

### MyId:

MyId is used to validate the user is a Disney employee. MyId also ties into Keystone to find if the user has elevated privileges. Any Disney Cast Member with a valid hub id can login and perform the basic functions of the application. i.e. putaway, bin inquiry, item inquiry, replenishment. Users who have been added to keystone and given GlobalAdmin or LocalAdmin role for the application will have additional functions. The super users are in charge of managing the Cast Members who will be added as GlobalAdmin or LocalAdmin within Keystone.

**Third Shift**

1. Manual Replenishment Functionality

**LeadCoordinator Elevated Privileges:**

1. Assign and Unassign items for other users
2. Reset Options
   1. Fill list
3. ASN Details

**LocalAdmin Elevated Privileges:**

1. Bulk Clear items
2. Reset Options
   1. Store nightly reset
   2. Fill list
3. Store Thresholds
4. Store Exclusions

**GlobalAdmin Elevated Privileges:**

1. Global Thresholds
2. Store Settings

# How To:

## How to validate application health

### Verify Servers:

1. Verify you can login to the **pride-svc** and **pride-ui** servers, with a tool such as putty, for the environment in question.
   1. [Production](#_Production:)
   2. [Disaster Recovery](#_Disaster_Recovery:)
   3. [Stage](#_Stage:)
   4. [Latest](#_Latest:)
   5. After login run the following command line command
      1. df –h
      2. This will show all mounts on the system
      3. Verify there is a “Mount on” the following locations
         1. /opt/apps/WDPRApps
         2. /var/opt/apps/WDPRApps
         3. /var/log
      4. Other mounts should display on the server but are not required for the pride application to function
      5. If these mounts do not exist you should open an INC to the SA team letting them know what mount point is missing.
         1. [Application SA assignment group](#_Service_Manager_Support)
         2. Join #wdpr-monitoring slack channel requesting help from an iMod SA and provide the incident number.
      6. Once the servers and mounts are verified continue to verify pride-svc.

### Verify pride-svc:

1. Verify the tomcat instances are up and running for pride-svc by calling the health check for the environment in question. Server specific healthcheck should be performed on the live instance.
   1. Loadbalancer health check
      1. <https://pride-svc.wdpr.disney.com:8443/pride-svc/deephealthcheck>
   2. Server specific health check (find live instance)
      1. <https://pride-svc.wdpr.disney.com:8443//dashboard>
      2. If "live\_instance": "a"
         1. <https://fldcvpsla17362.wdw.disney.com:8042/pride-svc/deephealthcheck>
         2. <https://fldcvpsla17363.wdw.disney.com:8042/pride-svc/deephealthcheck>
      3. If "live\_instance": "b"
         1. <https://fldcvpsla17362.wdw.disney.com:9042/pride-svc/deephealthcheck>
         2. <https://fldcvpsla17363.wdw.disney.com:9042/pride-svc/deephealthcheck>
      4. Response should be status OK with a 200 Http response code and a successful connection to prideDbConnection and simbaDbConnection
         1. {"status":"OK","httpStatusCode":"200","results":{"prideDbConnection":"success","simbaDbConnection":"success","environment":"PRD","tomcatApplication":"tomcatA2"}}
      5. If the response comes back with status=”WARN” then a connection to the database is not working.
         1. Verify which connection is not successful based on the response from the service.
            1. Verify that DB is up and running.

prideDbConnection connects to WD\_PICKLIST schema

simbDBConnection connects to SIM schema

See database connection information in the [environment](#_Installed_Environments:) section above.

* + - * 1. If there is an issue with the database open an INC to the Oracle DBA team to fix the [database](#_Production:) in question.

[Oracle DBA assignment](#_Service_Manager_Support)

Join #wdpr-monitoring slack channel requesting help from an oracle DBA DBO

* + 1. If no response comes back then create an INC to middleware requesting they start up tomcat on servers fldcvpsla1736[2-3].
       1. [Middleware assignment group](#_Service_Manager_Support)
       2. Join #wdpr-monitoring slack channel requesting help from middleware and provide the incident number.
    2. Once service response and database connections are verified continue to verify pride-ui

### Verify pride-ui:

1. Verify the nodejs instances are up and running for pride-ui by logging into the pride-ui.
   1. <https://pride.wdpr.disney.com/pride>
   2. If you are prompted, login and verify the store selection displays for the UI.
   3. If you are not promoted to login and receive a blank page or page not found open an INC to middleware requesting they start up nodejs on servers fldcvpsla1970[3-4].
      1. [Middleware assignment group](#_Service_Manager_Support)
      2. Join #wdpr-monitoring slack channel requesting help from middleware and provide the incident number.
   4. Once the pride-ui is validated continue to verify pride-batch.

### Verify pride-batch:

1. Login to [UC4](#_Production:) and verify the [batches](#_UC4_Batches:) are running.
   1. In Explorer view navigate to SIMBA>JOBP>PRIDE and verify that JOBP.SIMBA\_PRIDE.NIGHTLY\_REFRESH is setup and running every morning based on its configuration in UC4
   2. In Explorer view navigate to SIMBA>JOBS>PRIDE and verify that JOBS.SIMBA\_PRIDE.STORE\_DAILY\_RESET is setup and running on the hour based on its configuration in UC4
   3. In Explorer view navigate to SIMBA>JOBS>PRIDE and verify that JOBS.SIMBA\_PRIDE.SERVER\_LOG\_PURGE is setup and running on the weekly based on its configuration in UC4
   4. In Explorer view navigate to SIMBA>JOBS>PRIDE and verify that JOBS.SIMBA\_PRIDE.BATCH\_LOG\_PURGE\_A is setup and running on the weekly based on its configuration in UC4
   5. In Explorer view navigate to SIMBA>JOBS>PRIDE and verify that JOBS.SIMBA\_PRIDE.BATCH\_LOG\_PURGE\_B is setup and running on the weekly based on its configuration in UC4
   6. In Explorer view navigate to SIMBA>JOBS>PRIDE and verify that JOBS.SIMBA\_PRIDE.PICKLIST\_LOAD is setup and running about every 5 minutes based on its configuration in UC4
   7. Once all batches are validated the application can be confirmed up and running as expected.

## How to shut down and restart the Production Pride Applications

1. Shutdown and restarts can be requested with a service now request to the middleware team.

## How to change the DB configurations for pride-svc

DB configurations live within the application. Each environment has a resources folder which contains the environment specific connection information to the database as well as any external services (i.e. POS).

1. src/main/resources
   1. select the environment folder to be updated (DEV, DR, LOCAL, PRD, STG)
      1. update the database connection information in pride-svc.properties.json
      2. deploy the application with the updated connection properties
      3. **Note:** passwords are encrypted using jasypt as to not store the password in source control. The jasypt passphrase resides on the application server in a restricted folder. This passphrase is used to decrypt the password and allow the application connection into the database.
         1. Jasypt passphrase file location: /opt/middleware/jasypt/jasypt.properties
         2. Jasypt online documentation: <http://www.jasypt.org/>

## How to change the Matra connection information

1. Update the HOST\_NAME on the POS\_SERVER table for the POS server in question.

## How to change the Appetize connection information

1. pride-svc: Update the pride-svc.properties.json within the environment folder needing the connection update.
2. Checkin the code to github
3. Build and deploy the code to the environment through Jenkins.

## ~~How to generate a new Scandit License Key and Deploy (No longer using Scandit but left it the documentation in case we decide to use them in the future).~~

1. ~~Login to Scandit~~ 
   1. [~~https://ssl.scandit.com/main~~](https://ssl.scandit.com/main)
2. ~~Click on “License Keys” in the left hand navigation~~
3. ~~Click “Create Web SDK License Key”~~
4. ~~Enter a description and the domain for the key~~
   1. ~~i.e. Description: Pride Application~~
   2. ~~i.e. Domain Name: \*.disney.com~~
5. ~~Click “Create License Key”~~
6. ~~On the License Keys Page click on the newly created license Key column to display the entire key.~~
7. ~~Copy the license key~~
8. ~~Open pride-ui in eclipse or your favorite IDE.~~
9. ~~Open the environment specific file for each environment that needs the license key change.~~
   1. ~~/pride-ui/src/static/config/~~*~~{environment}~~*
10. ~~Replace the value of the SCANDIT\_KEY: variable with the copied Scandit license key.~~
11. ~~Push the changes to github~~
12. ~~Follow standard deployment in Jenkins to deploy the application to each environment.~~
    1. ~~Note: latest environment will auto deploy when code is merged into the develop branch.~~

## How to generate a new DataSymbol License Key and Deploy

1. Go to datasymbol’s website
   1. [https://datasymbol.com/order/order-barcode-reader-software.html](https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdatasymbol.com%2Forder%2Forder-barcode-reader-software.html&data=02%7C01%7CMelanie.Roehl%40disney.com%7Cb3363d0f9a644722bda808d6678fa208%7C56b731a8a2ac4c32bf6b616810e913c6%7C1%7C0%7C636810263918960791&sdata=H4TUwLWkxpUUBNFlBqg1eaCMYY5RUof3DqZRISSRYM8%3D&reserved=0)
2. Click on the Web SDK image to see the pricing for the Web SDK
3. Select the 1 year subscription radio button
4. Enter 3 for the Domain Name License Quantity
5. Select the Linear Professional checkbox
6. Click Buy Now

**License Details**

**Reader Type:** Web SDK

**Barcode Type:** Linear

**Edition:** Professional

**Domain License Quantity:** 3

1. After purchase the licenses DataSymbol should provide the ability to generate the license keys OR they may generate the license keys for you. The 3 license keys should be generated with the following domain names:
   1. pride-latest.wdpr.disney.com
   2. pride-stage.wdpr.disney.com
   3. pride.wdpr.disney.com
2. Update the keys in PRIDE-UI and deploy
   1. Open /src/static/config/(DEV.js/STG.js/PRD.js)
   2. Update the datasymbol key (DATASYMBOL\_KEY) in all of the files to correspond with the key received for each environment in the step above
   3. Pride-latest = DEV.js
   4. Pride-stage = STG.js
   5. Pride = PRD.js
3. Build and deploy to each environment thru CICD pipeline for pride-ui

## How to update a store to move to appetize

1. Request the appetize venue from the business.
2. Use the pride service to pull back the store details with an Https GET
   1. [https://pride-svc.wdpr.disney.com:8443/pride-svc/storenumber/{storeNumberHere}?storeType=SIMBA](https://pride-svc.wdpr.disney.com:8443/pride-svc/storenumber/%7bstoreNumberHere%7d?storeType=SIMBA)
   2. Service response example

{

"createDate": "2018-06-26",

"createUser": null,

"updateDate": "2019-06-06",

"updateUser": "KELLT072",

"id": 8,

"storeName": "(DOLPHIN) DISNEY SUNDRIES AND GIFTS",

"simbaStore": 13805957,

"posStore": 5957,

"posServerId": 29,

"picklistExclude": false,

"smartStockExclude": true,

"legacyPos": true,

"activeDate": "2018-06-26",

"simbaChainName": "WALT DISNEY WORLD",

"lastResetTime": 1559894481502,

"dailyResetTime": "04:00",

"fillListStatus": "N",

"fillListGenerated": 1559894482367,

"venue": null

}

1. Use pride service to update the store details and convert it to an Appetize store with an Https PUT
   1. <https://pride-svc.wdpr.disney.com:8443/pride-svc/store/configureSetup/>
      1. Request Headers
         1. Content-Type “application/json”
      2. Request Body
         1. Use the id from the request above as the storeId
         2. Set legacyPos to false
         3. Enter the venue received from the business as a string
         4. Enter your 8 character userId as user
         5. Example below

{

"storeId": 8,

"legacyPos": false,

"venue": "1507",

"user": “*YOUR\_USERID*"

}

1. Verify the update was successful by making a second call the service to retrieve the store details
   1. [https://pride-svc.wdpr.disney.com:8443/pride-svc/storenumber/{storeNumberHere}?storeType=SIMBA](https://pride-svc.wdpr.disney.com:8443/pride-svc/storenumber/%7bstoreNumberHere%7d?storeType=SIMBA)
   2. Service response example

{

"createDate": "2018-06-26",

"createUser": null,

"updateDate": "2019-06-06",

"updateUser": "KELLT072",

"id": 8,

"storeName": "(DOLPHIN) DISNEY SUNDRIES AND GIFTS",

"simbaStore": 13805957,

"posStore": 5957,

"posServerId": 29,

"picklistExclude": false,

"smartStockExclude": true,

"legacyPos": false,

"activeDate": "2018-06-26",

"simbaChainName": "WALT DISNEY WORLD",

"lastResetTime": 1559894481502,

"dailyResetTime": "04:00",

"fillListStatus": "N",

"fillListGenerated": 1559894482367,

"venue": “1507”

}

1. The store is now setup for appetize and will start pulling sales for replenishment

## How re-enroll a mobile device to an AirWatch profile

1. From the AirWatch launcher menu click on the admin icon
2. Login as admin
3. Exit the AirWatch launcher
4. Navigate to Factory Reset the device within the android settings
5. After reset is complete tap 7x anywhere in the whitespace on the screen
6. Scan the QR code



1. Set a 4 digit pin on the device
   1. 1971
2. Make sure the certificate has been applied on the device.
   1. If it is not properly applied then you will get a message while logging into the application in the browser that indicates that the site in unsecure/unsafe.
3. Test scanning within the Pride application.

# Troubleshooting:

## Store selector not showing

The main causes for the store selector not to show after logging into the application are as follows:

1. The UI cannot connect to the services because it is not available.
   1. To fix this make sure the pride-svc are up and available. A quick test can be run by calling the healthcheck service or the deephealthcheck service. The healthcheck will validate the application is up and running. The deephealthcheck will also validate that the connections to the database are working. Below are the links to the production healthchecks
      1. <https://pride-svc.wdpr.disney.com:8443/pride-svc/healthcheck>
      2. <https://pride-svc.wdpr.disney.com:8443/pride-svc/deephealthcheck>
2. The UI cannot connect to the services because the device using the application does not have the proper certificates installed.
   1. Check the browsers address bar to see if a Green Lock appears at the start of the url.
      1. If the lock is not green then the certificates are not installed properly on the device.
      2. Work with the AirWatch team to get the certificate properly installed on the device. The certificate needed is the Disney Root CA cert. The AirWatch team has this certificate and can send it to the device so that it can be imported and used.

## Replenishment List is showing a broken image for the item

This is caused when the scene 7 image that is setup in SIMBA is invalid. To fix this SIMBA must be updated to have a valid image link in scene 7 to display the image or the image must be removed from SIMBA to display the background castle.

**Note:** The image will not be available in PRIDE until the day after it is updated in SIMBA because we only pull updates from SIMBA 1x/day in the morning.

## Replenishment List is not showing any items or a particular item

The main causes for the replenishment list to be empty are as follows:

1. No items have met the thresholds for the store for the day
2. The store is setup for Smart Stockroom and no items are in bins.
   1. If a store is enabled for both replenishment list and smart stock room then an item must be in a bin before it will show in the replenishment list.
3. The batch to pull from the POS to get store sales is not updating the replenishment list. Verify the PICKLIST\_LOAD batch is running successfully in Atomic.
4. A store is setup for in PRIDE for the incorrect Item sales source.
   1. Item sales are pulled from either Matra or Appetize to determine if sales have met the threshold to show on the replenishment list. If a store has been setup to pull from Matra but it is actually an Appetize store then there will be no sales in Matra to pull. The same applies if the store is actually a Matra store but it has been configured to be an Appetize store.
   2. Update the STORE table fixing the LEGACY\_POS flag.
   3. The LEGACY\_POS is Matra so LEGACY\_POS=true (1) = Matra
   4. The LEGACY\_POS=false (0) = Appetize
5. The store thresholds are not set properly to allow items to show on the replenishment list.
   1. Items do not show on the replenishment list until an items sales meet a threshold.
   2. Threshold types are % of presentation quantity or sales quantity.
   3. Thresholds are configured globally on the DEPARTMENT table but can be overridden by each store.
   4. Thresholds can be overridden at the department, class, subclass or item level.
   5. A store item can only be overridden at one level at one time. The most granular classification takes precedent.
      1. i.e. If there is already a Class level threshold applied to a Store Item and a Department level threshold request is submitted, the Department level threshold will not overwrite the Class level threshold that already exists for that store item.
6. The item has been excluded from the replenishment list
   1. Items can be excluded for each individual store.
   2. Item exclusions can be at the department, class, subclass or item level.
      1. i.e. A class level exclusion will exclude all items in that class from the replenishment list for that particular store.

## I am unable to putaway an item

The main cause for this is that the item is not ranged to the store.

## I am not able to find an item in the item inquiry

The main cause for this is that the item is not ranged to the store.

## I am not able to see locations in the item inquiry

The main causes for locations not to show on the item inquiry screen are as follows:

1. The store may not be setup for smart stockroom.
   1. This can be changed on the Store Setup screen
   2. Update STORE table SMART\_STOCK\_ROOM\_EXCLUDE = false
2. The item has never been putaway in the application so we don’t know what bins contain the item.
   1. Use the putaway screen to scan the item and the bin to associate the item to the bin.

## Scanning is not working for some devices

This can happen when there is an issue applying an AirWatch update or something on the device profile has been corrupted. Refer to the how to section to re-enroll the devices that are having an issue into the AirWatch profile.

## Scanning is not working for all devices

The main causes for the scanning to stop working are as follows:

1. The certificates are not imported on the devices.
   1. Check the browsers address bar to see if a Green Lock appear at the start of the url.
      1. If the lock is not green then the certificates are not installed properly on the device.
      2. Work with the AirWatch team to get the certificate properly installed on the device. The certificate needed is the Disney Root CA cert. The AirWatch team has this certificate and can send it to the device so that it can be imported and used.

**If using Datasymbol for scanning**

1. The datasymbol license may have expired. Refer to the How To section on how to regenerate the license key.

**If using Scandit for scanning**

1. There are no more Scandit licenses available.
   1. Work with the devices team and Scandit to get additional licenses.
2. The Scandit license has expired.
   1. Scandit licenses are annual and per device. Work with the devices team and Scandit to renew the licenses.

# Disaster Recovery:

PRIDE is included in the Master Disaster recovery plan for SIMBA. The Disaster recovery plan can be located at:  
<https://corp.sharepoint.disney.com/sites/msrp/SimbaIT/Project%20Directory%20New%20version/Disaster%20Recovery%20Information/Disaster%20Recovery%20Plans/SIMBA%20Disaster%20Recovery%20Plan%20v5.docx>

# Appendix:

# Data Dictionary Pride DB Tables

## STORE\_ITEM\_BIN\_XREF

A table containing the link between a store item and its bin location.

### IMPORTANT COLUMNS

1. BIN\_ID – The BIN.ID value
2. STORE\_ITEM\_ID – The STORE\_ITEM.ID value
3. STATUS – The action performed on the STORE\_ITEM/BIN combination
   1. PUTAWAY – The Item was putaway using the putaway screen
   2. REMOVED – The Item was removed in the Bin Inquiry screen. Item only removed (trash can icon)
   3. CLEARED – The Item was removed in the Bin Inquiry by clearing the bin
   4. SHORTPICK – The item was removed from the bin due to a short pick (RECOMMENDED\_QTY > PICK\_QTY) through the replenishment list.

## STORE\_ITEM\_BIN\_TRANS\_LOG

A log record is created in this table any time a STORE\_ITEM\_BIN\_XREF record is added, updated or deleted. If it does already exist then the record is updated with the UPDATE\_DATE, UPDATE\_USER and STATUS. XREF can be added or updated in the PUTAWAY and Bin Inquiry screens and can be deleted in the Bin Inquiry screen.

### IMPORTANT COLUMNS

Refer to STORE\_ITEM\_BIN\_XREF for important columns

## BIN

A table containing the bin details for stores with smart stockroom enabled.

### IMPORTANT COLUMNS

1. CODE – The barcode for the bin in the stockroom
2. STOCKROOM – The stockroom the bin is located in based on the BIN.CODE. The barcodes are generated using the stockroom as the first character of the BIN.CODE. This column is for increased database performance when retrieving stockroom locations
3. STORE\_ID – The id for the store that the bin is located in.

Note: STORE\_ID/CODE is unique. There can only be one BIN.CODE per STORE

## BIN\_TRANS\_LOG

A log record is created in this table any time a BIN record is added or deleted. BIN records are added in the PUTAWAY screen if no CODE for that STORE\_ID exists. If the STORE\_ID/CODE combination already exists and another item is added to the BIN then no changes are made to the BIN and no log record is created in the BIN\_TRANS\_LOG.

Note: BINs are only CREATED and DELETED so the UPDATE/CREATE\_DATE/USER will always be the same.

### IMPORTANT COLUMNS

Refer to BIN table for important columns

## STORE\_ITEM

A table containing the store specific details for the item pulled from SIM in a daily store item batch.

### IMPORTANT COLUMNS

1. STORE\_ID – The STORE.ID value
2. ITEM\_ID – The ITEM.ID value
3. DAILY\_PICK\_CONF\_QTY – The running daily total of completed PICKLIST records for the store item. This running total is the sum of the PICKLIST.PICK\_QTY for PICKLIST records that are COMPLETE or DECLINED plus the PICKLIST.RECOMMENDED\_QTY for CLEARED records. This column is reset daily based on the STORE.DAILY\_RESET\_TIME. For stores open after midnight the DAILY\_PICK\_CONF\_QTY is set to the value pulled from POS sales before STORE.DAILY\_RESET\_TIME.
4. PRES\_QTY – The store item floor presentation quantity pulled from SIM every morning during the daily store item batch.
5. OUT\_OF\_STOCK – Boolean (1/0) flag indicating if the store item has been short picked (RECOMMENDED\_QTY > PICK\_QTY) for the day. If this flag is set TRUE (1) then the store item will not show on the replenishment list until the next business day. This column is reset daily based on the STORE.DAILY\_RESET\_TIME.
6. PICKLIST\_EXCLUDE\_LVL – Hierarchical level in which that store item has been excluded on
   1. DEPARTMENT – This item falls under a department that has been excluded
   2. CLASS – This item falls under a class that has been excluded
   3. SUBCLASS – This item falls under a subclass that has been excluded
   4. ITEM – This item has been excluded
7. DELETE\_DATE – The date at which the store item will be deleted from the database. This date is determined based on the store item daily batch. If the item is not found in SIM but is present in PRIDE then the item is no longer ranged to the store. We provide a 30 day grace period for these items to allow for liquidation of the stock on hand before we remove it from our tables.
8. PQ\_OVERRIDE\_DATE – The date the items presentation quantity override will end. A store can chose to override a presentation quantity for up to 14 days. This is not updated in SIM and is only used for replenishment threshold calculations. Once the override date is in the past the presentation quantity will reset back to the value from SIM during the stores daily reset. If the store item is deleted from SIM or is already flagged for deletion from SIM then the presentation quantity will stay at the overridden value until it is deleted or updated to a new value from the UI.
9. PQ\_OVERRIDE\_USER – The last user to override the presentation quantity for the store item.
10. UNIT\_PRICE – A snapshot of the store item unit price from SIM at the time the STORE\_ITEM batch is run.

## STORE

A table containing the store specific details pulled from SIM in a daily store batch.

### IMPORTANT COLUMNS

1. STORE\_NAME – The name of the store pulled from SIM
2. SIMBA\_STORE – The store number pulled from SIM
3. POS\_STORE – The last 4 digits of the SIMBA\_STORE which matches back to the number used in the POS system (MATRA/APPETIZE)
4. POS\_SERVER\_ID – The POS\_SERVER.ID value to map the store back to MATRA to pull stores daily sales
5. PICKLIST\_EXCLUDE – A flag (1/0) indicating if this store is setup to be excluded from the replenishment list features. (Item Inquiry, Replenishment List)
6. SMART\_STOCK\_EXCLUDE – A flag (1/0) indicating if the store is setup to be excluded from the smart stockroom features. (Putaway, Bin Inquiry, Item Inquiry)
7. LEGACY\_POS – A flag indicating if the sales to pull for the store should come from MATRA (LEGACY\_POS=1) or APPETIZE (LEGACY\_POS=0).
8. SIMBA\_CHAIN\_NAME – The stores region (WALT DISNEY WORLD or DISNEYLAND) pulled from SIM
9. LAST\_RESET\_TIME – A date time indicating when the store was last reset for replenishment totals. This is updated when the store nightly reset batch runs based on the STORE.DAILY\_RESET\_TIME or reset through the UI.
10. DAILY\_RESET\_TIME – The time at which the store item values and replenishment list will be reset for the next day. All replenishment items are cleared from the store and the STORE\_ITEM.OUT\_OF\_STOCK and STORE\_ITEM.DAILY\_PICK\_CONF\_QTY values are reset.
11. FILL\_LIST\_STATUS – A flag indicating if the fill list batch job is currently running
12. FILL\_LIST\_GENERATED – The last time the fill list was run for the store.

## STORE\_TRANS\_LOG

A log record is created in this table any time a STORE record is added, updated or deleted. This is used to track changes made to the store.

### IMPORTANT COLUMNS

Refer to STORE table for important columns

## ITEM

A table containing item specific details pulled from SIM in a daily item batch.

### IMPORTANT COLUMNS

1. SIMBA\_ITEM – The item number pulled from SIM
2. ITEM\_DESC – The item description pulled from SIM
3. DEPT\_ID – The DEPT.ID value. The link back to the department table to provide department details.
4. SIMBA\_SUBCLASS – The item subclass number pulled from SIM
5. SIMBA\_SUBCLASS\_NAME – The item subclass name pulled from SIM
6. SIMBA\_CLASS – The item class number pulled from SIM
7. SIMBA\_CLASS\_NAME – The item class name pulled from SIM
8. IMAGE\_URL – The image url pulled from SIM to provide the location of the image for the item
9. PARENT\_REF\_ID – The Simba parent reference id to link items together. This link will help us determine what items show in the related items link on the UI. i.e. A Mickey t-shirt size S would have the same parent ref id of the same t-shirt in a different size: M, L, XL, etc.

## ITEM\_CODE

A table containing external vendors UPCs. This table will cross reference the SIMBA\_ITEM to a vendors UPC.

### IMPORTANT COLUMNS

1. ITEM\_ID – The ITEM.ID value to map back to the SIMBA\_ITEM
2. UPC\_CODE – The vendors UPC code assigned to the item

## STORE\_ITEM\_THRESHOLD

A table containing the overridden thresholds for a store item. Each Department has a default threshold for all items in that department. The STORE\_ITEM\_THRESHOLD table allows a store to override the default threshold for an item at the DEPARTMENT, CLASS, SUBCLASS or item level. There will only be 1 STORE\_ITEM\_THRESHOLD for each STORE\_ITEM. A STORE\_ITEM can only be overridden at 1 level at a time. The override hierarchy from top to bottom is DEPARTMENT, CLASS, SUBCLASS and ITEM. STORE\_ITEM\_THRESHOLDs will override at the lowest level provided. i.e. If a STORE\_ITEM is overridden at the DEPARTMENT level and the CLASS for the item is subsequently overridden then the STORE\_ITEM will update its THRESHOLD\_LEVEL to class. The reverse, however, is not true.

### IMPORTANT COLUMNS

1. CRITICAL\_PRES\_THRESHOLD – The presentation quantity percentage to be used to flag a replenishment item as critical. The percentage is the percent left on the sales floor not the percent sold. The percent is in whole numbers i.e. 80 not .80 for 80 percent. The percent is based on STORE\_ITEM.PRES\_QTY. i.e. STORE\_ITEM.PRES\_QTY = 10 with a CRITICAL\_PRES\_THRESHOLD = 20% means the item will show up on replenishment list as CRITICAL when 8 units are sold (20% of the items are still on the sales floor).
2. WARNING\_PRES\_THRESHOLD - The presentation quantity percentage to be used to flag a replenishment item as warning. The percentage is the percent left on the sales floor not the percent sold. The percent is in whole numbers i.e. 80 not .80 for 80 percent. The percent is based on STORE\_ITEM.PRES\_QTY. i.e. STORE\_ITEM.PRES\_QTY = 10 with a WARNING\_PRES\_THRESHOLD = 80% means the item will show up on replenishment list as WARNING when 2 units are sold (80% of the items are still on the sales floor).
3. CRITICAL\_SALES\_THRESHOLD – The number of units sold to flag a replenishment item as critical.
4. WARNING\_SALES\_THRESHOLD – The number of units sold to flag a replenishment item as warning.
5. THRESHOLD\_TYPE – The type of threshold to use when calculating if an item should show on replenishment.
   1. SALES\_QTY – Will use the CRITICAL/WARNING\_SALES\_THRESHOLD to determine if an item should show on replenishment list
   2. PRES\_QTY – Will use the CRITICAL/WARNING\_PRES\_THRESHOLD to calculate the units sold vs the presentation quantity to determine if the item should show on replenishment list
6. STORE\_ITEM\_ID – The STORE\_ITEM.ID value that the threshold is overriding.
7. THRESHOLD\_LEVEL – Indicates at what level the store items was overridden
   1. DEPARTMENT – All items in the department are overridden at this level
   2. CLASS – All items in the class are overridden at this level
   3. SUBCLASS – All items in the subclass are overridden at this level
   4. ITEM – The item is overridden at this level

## STORE\_ITEM\_THRESHOLD\_TRANS\_LOG

A log record is created in this table any time a STORE\_ITEM\_THRESHOLD record is added, updated or deleted. This is used to track changes made to the store item threshold.

### IMPORTANT COLUMNS

Refer to STORE\_ITEM\_THRESHOLD for important columns

## DEPARTMENT

A table containing department details pulled from SIM in a department batch

### IMPORTANT COLUMNS

1. DEPT\_NUMBER – The department number pulled from SIM
2. DEPT\_NAME – The department name pulled from SIM
3. CRITICAL\_PRES\_THRESHOLD – The default threshold given to all items in the department. This can be overridden with a STORE\_ITEM\_THRESHOLD. The presentation quantity percentage to be used to flag a replenishment item as critical. The percentage is the percent left on the sales floor not the percent sold. The percent is in whole numbers i.e. 80 not .80 for 80 percent. The percent is based on STORE\_ITEM.PRES\_QTY. i.e. STORE\_ITEM.PRES\_QTY = 10 with a CRITICAL\_PRES\_THRESHOLD = 20% means the item will show up on replenishment list as CRITICAL when 8 units are sold (20% of the items are still on the sales floor).
4. WARNING\_PRES\_THRESHOLD - The default threshold given to all items in the department. This can be overridden with a STORE\_ITEM\_THRESHOLD. The presentation quantity percentage to be used to flag a replenishment item as warning. The percentage is the percent left on the sales floor not the percent sold. The percent is in whole numbers i.e. 80 not .80 for 80 percent. The percent is based on STORE\_ITEM.PRES\_QTY. i.e. STORE\_ITEM.PRES\_QTY = 10 with a WARNING\_PRES\_THRESHOLD = 80% means the item will show up on replenishment list as WARNING when 2 units are sold (80% of the items are still on the sales floor).
5. CRITICAL\_SALES\_THRESHOLD – The default threshold given to all items in the department. This can be overridden with a STORE\_ITEM\_THRESHOLD. The number of units sold to flag a replenishment item as critical.
6. WARNING\_SALES\_THRESHOLD – The default threshold given to all items in the department. This can be overridden with a STORE\_ITEM\_THRESHOLD. The number of units sold to flag a replenishment item as warning.
7. THRESHOLD\_TYPE – The type of threshold to use when calculating if an item should show on replenishment.

## DEPARTMENT\_TRANS\_LOG

A log record is created in this table any time a DEPARTMENT record is added, updated or deleted. This is used to track changes made to the departments default threshold values and type.

### IMPORTANT COLUMNS

Refer to the DEPARTMENT table for Important Columns

## PICKLIST

A table containing the store items that need replenishment based on the threshold for the store item. This table is updated by the picklist batch when store item sales change (~5 min) or through the Replenishment List screen. If an item has been completed (COMPLETE, CLEARED, DECLINED, DUPLICATE, IGNORED) then a new record will be created when the threshold is met again. If an item has not been completed and there is additional information then the existing record will be updated.

### IMPORTANT COLUMNS

1. STORE\_ITEM\_ID – The STORE\_ITEM.ID value. The store item that needs to be replenished to the sales floor.
2. STATUS – The status of the item on replenishment.
   1. OPEN – The item is available for any cast member to replenish
   2. ASSIGNED – The item is assigned to a cast member and only available for that cast member to replenish.
   3. COMPLETE – The item was picked and is no longer available to work.
   4. CLEARED – The item was not picked and removed from the replenishment list.
   5. DECLINED – If PICK\_QTY = 0 then there were no items in a bin or the item was not found. If the PICK\_QTY > 0 then there were items found but less than the RECOMMENDED\_QTY and that item is not located in any other bins for that store
   6. DUPLICATE – This is created by the system only in the case of bad data, having move then 1 STORE\_ITEM\_ID in OPEN or ASSIGNED status at the same time. The picklist batch is self-healing and will merge the 2 DUPLICATE records together with the correct recommended quantity on its next run (~5 min). DUPLICATE is the original record that is ignored and a new record created with the appropriate information. DUPLICATE records should be ignored for reporting purposes.
   7. IGNORED – The item was on replenishment for the users to replenish, however, it was not replenished before the store was reset for the next day. This is useful to see what items were not replenished by the user but probably should have been.
3. RECOMMENDED\_QTY – The quantity determined by the system that is needed to be replenished to the store sales floor. PICKLIST.QTY\_SOLD - STORE\_ITEM.DAILY\_PICK\_CONF\_QTY
4. PICK\_QTY – The quantity actually picked by the cast member to replenish to the store sales floor
5. ASSIGN\_DATE – The date the user was assigned the picklist item
6. ASSIGNED\_USER\_ID – The user id that was assigned the item
7. COMPLETE\_DATE – The date the item was complete. Complete is defined as one of the following: COMPLETE, CLEARED, DECLINED, DUPLICATE
8. QTY\_SOLD – The total number of items sold for the day
9. CRITICAL\_LEVEL – The priority of the item to be replenished to the sales floor based on thresholds.
   1. WARNING – A level showing the item has sold but is not in danger of running out on the sales floor based on the threshold configured for the item.
   2. CRITICAL – A level showing that the item is reaching a point at which it may be sold out on the sales floor and needs replenishment to keep it from selling out.
   3. FILL- A user can generate a fill list which comprises of items that were sold, but didn’t meet the threshold. This is usually done at the beginning or end of the day to fully stock the sales floor. This category represents those items.
   4. MANUAL – An item that was added by the users manually. This is typically used when an item needs to be replenished but hasn’t hit a threshold or it has hit a threshold but more than the recommended quantity needs to be picked. This is usually needed at the end of the day to fill the floor for items that are missing that didn’t have a sale. i.e. Lost or stolen.
10. BIN\_LOCATION\_CODE – The BIN.CODE value from which the item was picked from.
11. PICKLIST\_REF\_ID – The PICKLIST.ID value. If smart stockroom is enabled and an item is in multiple locations and the item is short picked (RECOMMENDED\_QTY > PICK\_QTY) then a new picklist record will be created and the reference to the new picklist record will be available in this column.

## PICKLIST\_TRANS\_LOG

A log record is created in this table any time a PICKLIST record is created, updated or deleted. PICKLIST records are created by the system from a batch when a threshold is met. Records are updated when a user assigns, unassigns, declines, short picks, clears, or picks a record from the Replenishment list.

### IMPORTANT COLUMNS

Refer to PICKLIST table for important columns.

## ITEM\_DELETES

A table containing a list of items to delete during the nightly item batch. This table will only have items while the item batch is running. The batch determines what items need to be deleted and it writes them to this table. After it has identified all of the items it will read from this table deleting all ITEMs and cascading deletes.

## POS\_SERVER

A table containing the information to connect to the MATRA (STORE.LEGACY\_POS=1) to pull sales data for the store. MATRA has different servers for different stores to pull the sales data from.

### IMPORTANT COLUMNS

1. SERVER\_NAME – The name given to the MATRA server
2. HOST\_NAME – The host URL for the MATRA server
3. STORE\_URL – The URL needed to pull store information
4. DEPARTMENT\_URL – The URL needed to pull department details
5. STORE\_PERIOD\_URL – The URL needed to pull the store period based on the date so that item movement can be pulled for the period.
6. ITEM\_MOVEMENT\_URL – The URL needed to pull the item movement for the store for the provided period.