ArgixDirectIntranetHome

Bearware Export

Design Document

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 05/24/06 | 1.0 | Created | Jim Heary |
| 03/17/08 | 1.1 | Updated | Jim Heary |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Overview 3

Business Context 3

Key Abstractions 4

Object States 4

Application Design 5

Overview 5

Design Model 5

Export Format 5

Export Fields 5

Export File- Sample 5

Deployment 6

Overview 6

Deployment Model 6

Runtime Configuration 7

Configuration File (app.config) 7

Database Configuration Table 7

File Structure 7

# Overview

The Bearware Export application (BearwareExport.exe) allows Argix terminals to export carton information from directly sorted freight into a file that can be imported into the Bearware System. From this information, [Bearware] clients can generate OS&D and POD information from original vendor BOL/manifest paperwork. The Bearware Export application fulfills the requirements of the business process activity Export Sorted Items shown below.

## Business Context



## Key Abstractions



## Object States

N/A

# Application Design

## Overview

## Design Model



# Deployment

## Overview

The Bearware Export application creates an export file that is imported into the Bearware System.

## Deployment Model



## Runtime Configuration

The runtime configurable application parameters are set in the application configuration file and in the Configuration table in the applications database. The following two sections show the run-time configuration required to test the application.

### Configuration File (app.config)

Open the application configuration file (i.e. BearwareExport.exe.config), go to the <configuration> section, and add/edit the values below:

<configuration>

<!-- Test configuration -->

<configSections>

<sectionGroup name="menu">

<section name="help" type="System.Configuration.NameValueSectionHandler" />

</sectionGroup>

</configSections>

<menu>

<help>

<add key="Contents..." value=" htm" />

<add key="Release Notes..." value=" " />

<add key="Issue Log..." value=" " />

</help>

</menu>

<appSettings>

<!-- Client configuration -->

<add key="UseWebSvc" value="false" />

<add key="EventLogMachineName" value="RGXDEV" />

<!-- Tsort.Data configuration -->

<add key="Tsort.Data.DataWS.DataAccess" value="http://RGXDEV/RGXDEVDB.TSORTLOCAL.WebSvc/DataAccess.asmx" />

<add key="SQLConnection" value="data source=RGXDEVDB;initial catalog=TSORTLOCAL;persist security info=False;user id=\*;password=\*;packet size=4096;Connect Timeout=30" />

</appSettings>

</configuration>

### Database Configuration Table

Open the local sort database (i.e. TSORTLOCAL on RGXDEVDB), goto the Configuration table, and add/edit the values below:

Application PCName Key Value Security

--------------- --------------- ------------------------- ------------------------- --------

Bearware Export Default DateDaysBack 60 1

Bearware Export Default MISPassword \*\*\*\*\*\*\*\* 1

Bearware Export Default TraceLevel 3 1

### File Structure

Bearware Export is installed into a deployment folder visible from an IIS web server as follows:

1. Create a virtual folder c:\inetpub\wwwroot\IndirectSort on a web server like RGXDEVDB using Windows Explorer.

2. Copy the following files into this folder:

BearwareExport.exe

BearwareExport.exe.config

Infragistics.Shared.v6.3.dll

Infragistics.Win.UltraWinGrid.v6.3.dll

Infragistics.Win.v6.3.dll

Microsoft.ApplicationBlocks.Data.dll

releasenotes.txt

Tsort.AppLogging.dll

Tsort.Controls.dll

Tsort.Shared.User.dll

3. Launch the Internet Information Services snap-in and validate that the virtual folder allows anonymous access. A web application does not need to be created for deployment.

# Appendix: RDS Export Format

## Export Files

Client File

filename = #########0.txt

contents = 1 Trailer record, 0+ Sorted Item records

Argix File

filename = #########0.txt

contents = 1 Trailer record, 0+ Sorted Item records

### Trailer Record

TPAYLES06JSBM00381786527 109873 DIGYI

TBLOCKBPPSODM0078233684A69A 0030108 BILTI

-======--===--------========--------====-

|| | | | | | | |

ab c d e f g h i

a) Record Type [Required; 1]

- Constant "T"

b) Company Code (See Company Code Table) [Max Length 6]

- ANNTAY = Ann Taylor; LTA = Argix

c) Scanner Number [Required; 2]

– Blank

d) Scanner UserID [Required; 3]

– Blank

e) Trip Number [Required; 8]

– Client file = sorted item vendorKey (first 8 positions)

- Argix file = sorted item TDSID (last 8 positions)

f) Trailer Number [8]

– Trailer# or blank

g) Seal Number [8]

– Seal# or blank

h) Carrier SCAC Code [4]

– Blank

i) Scanner File Type [Required; 1]

- I = Inbound Scan

### Sort Record

S29127133277 020021220094806123456

S000744232835 020040809084200

S1850200205252 020040809102100

S060440839041028135854050 020041102084439

S0000420018049386522806125 020061031082012

-==========================-========------======

|| || | |

ab cd e f

a) Barcode Origin [1]

– S = Scanned; M = Manual (always S for RDS Export application)

b) Barcode Stream [26]

- Client file = sorted item vendor item#

- Argix file = sorted item# (build 24 position: client#+44/88+store#+labelseq#+0)

c) Damage Code [1]

- sorted item damage code

d) Scan Date [8, YYYYMMDD]

- sorted item sort date

e) Scan Time [6, HHMMSS]

- sorted item end time

f) Override (Inbound) or From Store (Return, Transfer) [6]

– Blank