This page intentionally left blank
Change History

<table>
<thead>
<tr>
<th>Rev A</th>
<th>Initial version</th>
<th>October 2012</th>
</tr>
</thead>
</table>

This page intentionally left blank
Disclaimer
NOTICE TO ALL PERSONS RECEIVING THIS DOCUMENT:
The information in this document is subject to change without notice. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TransAct Technologies, Inc. ("TransAct"). This document is the property of and contains information that is both confidential and proprietary to TransAct. Recipient shall not disclose any portion of this document to any third party.

TRANSACT DOES NOT ASSUME ANY LIABILITY FOR DAMAGES INCURRED, DIRECTLY OR INDIRECTLY, FROM ANY ERRORS, OMISSIONS OR DISCREPANCIES IN THE INFORMATION CONTAINED IN THIS DOCUMENT.

TransAct cannot guarantee that changes in software and equipment made by other manufacturers, and referred to in this publication, do not affect the applicability of information in this publication.

Copyright
© 2012 TransAct Technologies, Inc. All rights reserved. Printed in USA
Part 100-12152, Revision Level A, October 2012

Trademarks
Some of the product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

TransAct, Ithaca, and Printrex are registered trademarks of TransAct Technologies, Inc.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe and the Adobe logo are trademarks of Adobe Systems, Inc.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change History</td>
<td>i</td>
</tr>
<tr>
<td>Disclaimer</td>
<td>iii</td>
</tr>
<tr>
<td>Copyright</td>
<td>iii</td>
</tr>
<tr>
<td>Trademarks</td>
<td></td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iv</td>
</tr>
<tr>
<td>About the Menu Editor</td>
<td>1</td>
</tr>
<tr>
<td>Who Should Read This Guide?</td>
<td>3</td>
</tr>
<tr>
<td>What Is Included in This Guide?</td>
<td>3</td>
</tr>
<tr>
<td>About the Food Safety Terminal Menu Editor</td>
<td>4</td>
</tr>
<tr>
<td>Using the Menu Editor</td>
<td>5</td>
</tr>
<tr>
<td>Overview of Application</td>
<td>7</td>
</tr>
<tr>
<td>Running the Application</td>
<td>7</td>
</tr>
<tr>
<td>Overview of Data</td>
<td>9</td>
</tr>
<tr>
<td>Overview of Data Files</td>
<td>11</td>
</tr>
<tr>
<td>Menu Data XML File</td>
<td>11</td>
</tr>
<tr>
<td>Layout Data XML File</td>
<td>12</td>
</tr>
<tr>
<td>Using the Menu Editor Application</td>
<td>13</td>
</tr>
<tr>
<td>Main Screen Functions</td>
<td>13</td>
</tr>
<tr>
<td>Items/Categories Screen Functions</td>
<td>14</td>
</tr>
<tr>
<td>Appendix A: Sample XML Files</td>
<td>21</td>
</tr>
<tr>
<td>Sample XML File for Food Menu data</td>
<td>21</td>
</tr>
<tr>
<td>Sample XML File for Food Label Layout</td>
<td>23</td>
</tr>
<tr>
<td>Index</td>
<td>27</td>
</tr>
</tbody>
</table>
Chapter 1

About the Menu Editor
This page intentionally left blank
Who Should Read This Guide?

This document provides information useful for end users who will maintain label configuration data for the Ithaca® 9700 Food Safety Terminal in their food service operations.

What Is Included in This Guide?

This Guide documents the operation of the Ithaca® 9700 Food Safety Terminal Menu Editor, a user-friendly PC-based application for maintaining the data files used to define label data for the Ithaca® 9700.

For any issues not covered in this guide, quality technical support is available on-line at www.transact-tech.com, or by telephone at 607-257-8910 in the United States. Consult our web site for further support information, including international sales and support offices.
About the Food Safety Terminal Menu Editor

The Ithaca® 9700 Food Safety Terminal is a compact, easy-to-use label terminal designed for use in food service environments, for applications such as printing food safety and nutrition labels for food items. Labels are edited and printed using a simple touch-screen menu. Details on the operation of this terminal are described in the *Ithaca® 9700 Food Safety Terminal Operators Guide*, also available from TransAct.

Formatted data files written using Extensible Markup Language (XML) are used to specify the following data controlled by these touch-screen menus:

- Categories of food or other items
- Specific items within these categories
- Label data for these items
- Layout data specifying the position of these data on printed labels

It is technically possible to modify these XML files directly, in either raw or spreadsheet format, to maintain these items. However, this requires programming expertise in the XML language, a human-readable language for document definition that is often complex and verbose. This approach also requires loading data files to the terminal for verification and troubleshooting.

By comparison, the Ithaca® 9700 Food Safety Terminal Menu Editor is a visual, easy-to-use Java application that allows you to read in an existing XML file, edit or modify its contents using simple graphical menus, and write out the modified file to subsequently load and use in your terminal. This Operators Guide details the use of this application.

**Note:** This application only supports editing of category, item, and item data fields. Layout data must be manually edited in XML, as described later in this document.
Chapter 2

Using the Menu Editor
Overview of Application

The Food Safety Terminal Menu Editor is a straightforward, purpose-built application that performs the following functions:

- Reads in two XML files for menu and layout definition
- Allows you to graphically edit categories, items, and item data within this menu file
- Output the revised XML files, which can then be loaded on your Ithaca® 9700 terminal

Once you have loaded the XML files for a project, you will be presented with a main screen showing categories of items at the bottom, the items themselves on the main screen, and navigation buttons, as shown in the figure below.

Running the Application

The Food Safety Terminal Menu Editor is a Java application that runs as an executable program and ships as an executable file with a name similar to TransactFST9700_xxx.jar

where "xxx" is the version number.

**Note:** To run this application, Java version 1.7.0 or later must be installed on the target system.

To run the application, either double click on the executable file, or the application may be launched by opening a command prompt and executing the command “java -jar <Path>TransactFST9700_xxx.jar” where <Path> is the path to the executable file on disk, and "xxx" is the supplied version number.

**Note:** This application is provided as a bundled .jar file and besides the Java runtime, requires no other packages to be present on the system.
Categories and items, described further below, can be edited by either clicking on them, or right-clicking and selecting the editing function. Other functions accessible by right-clicking specific categories include adding or deleting categories. Other functions accessible by right-clicking specific items include creating duplicate items from the one clicked and deleting items.
Overview of Data

The data edited by this program follows a hierarchical structure as follows:

- **Categories** define common types of items shown together on the screen of the Ithaca® 9700 terminal for label printing. These are most commonly categories of food items, such as breakfast items, condiments, or desserts.

- **Items** are the individual entities for which labels are printed, such as bagels or sandwiches.

- **Item Data** is the specific information printed on the label for this item, such as a sell-by date or nutritional information.

**Note:** The values of existing item data fields can be edited using this program. Definitions of item data fields themselves must be edited manually in the menu data XML file, and their position in printed labels is defined in the XML layout file, which must be also edited manually. These fields are rarely changed in daily restaurant usage.

Any number of categories, and items within these categories, can be defined. Individual food items can then be added to these categories and detailed information for label printing can be defined for the food items.

For example, the category "Soups" can contain different soup items such as vegetable soup or minestrone soup, while the category "Salad" can contain salad-based food items such as Greek salad or mixed salad. Each of these food items can have further associated information stored in the menu data XML file, as shown in the figure.
Salads
- Greek Salad
  • Ingredients
  • Shelf Life
  • Price
- Caesar Salad
  • Ingredients
  • Shelf Life
  • Price

Soups
- Vegetable Soup
  • Ingredients
  • Shelf Life
  • Price
- Minestrone Soup
  • Ingredients
  • Shelf Life
  • Price

Breakfast
- Muffins
  • Ingredients
  • Shelf Life
  • Price
- Bagels
  • Ingredients
  • Shelf Life
  • Price
Overview of Data Files

This program uses two XML files that define the terminal menu firmware, a menu data file and a layout file. Samples of both files are provided in Appendix A, annotated to show their format and contents.

Menu Data XML File

The menu data XML file includes all the information for food categories and food items.

A food category is the top component in the label hierarchy, and can be used to group together food items with similar information and layouts. Each category has a label layout associated with it, and this layout is contained in the label layout XML file (described below).

When a food item is defined a layout is used to display the fields and their position in the item label. The edit layout for a category is known as “MasterEditLayout”.

A food category includes the following information as part of its XML data:

- **Color** - The color for the category. All food items belonging to a category are displayed using the color value defined for the category.

- **Language** - The language to be used for category and item data display. This is not currently utilized in the Ithaca® 9700 and the language will always be set to “default”. This feature is implemented for future expansion to a “multiple” language database where there are duplicates of everything for each language supported. The system currently supports direct entry of Category and Item names in both a default language and an alternate language.

- **PrintTemplate** - A Transact label printer supports different pre-defined print layouts for food labels. This field sets the default layout for printing the items belonging to a given category.

- **Printer** - This sets the default printer to be used for printing label data for this category and its food items. This implementation varies for different customers to customer. The printer selection may also be associated with one to several, print buttons. In addition, a layout is associated with a print button.

The fields in the food item are defined either based on the fields defined under the tag “<SupportedLabelFields> </SupportedLabelFields>” in the menu data XML file or the layout fields defined under a layout for the item category in the label layout XML file.

**Note:** Food item fields must be manually defined in both the menu data and layout data XML files before they can be used to edit item information. Consult Appendix A for examples of how this field data is defined.
Layout Data XML File

This file contains information on label layouts for printing on the terminal and for editing on the screen. It normally has a default filename of "layout.xml".

Several possible layouts can be defined in the layout xml file. This file must be manually edited to modify or define new layouts. Layouts include all the information regarding the print layout for a food item. This allows the user to see on screen a representation of the format of the label that will be printed on the terminal.

Appendix A contains an annotated example of the format of the layout data XML file.
Using the Menu Editor Application

Main Screen Functions
The main screen, shown in the figure below, appears when the application is started.

![Main Screen](image)

This main screen provides three options to load, save and edit menu data information, described below.

**Note:** These three functions are also accessible via the File menu.

**Open Data From Disk**
This button first prompts the user to select an XML file containing the menu data. Next, if needed, the user is prompted to select an XML file containing the layout data. The XML file opened can be an Excel spreadsheet saved in XML format, or a MenuData file output from the Menu Editor application.

By default, the file “layouts.xml” is loaded with no user intervention if present in the same directory as the Menu Editor application. If no layouts.xml file is detected by the application, the user is prompted for a filename for the layout data.

After loading the selected menu data file with category and food item information, the user is presented with the Items/Categories screen described in the following section.

**Note:** The contents of the screen depend upon the actual content of the menu data file.
Save Data To Disk
This selection allows the user to save a loaded “menu data” file to the disk. A standard Windows Save File dialog allows the user to select the directory and the filename to save the loaded/edited XML file. The file can be saved with any name, however the file name must be “menudata.xml” when the file is loaded onto the Ithaca<sup>®</sup> 9700 Food Safety Terminal for use.

Continue Data Edit
The Continue Data Edit button allows the user to move from the start screen back to the editing screen to view and edit the currently loaded XML file.

Items/Categories Screen Functions

When the user loads a menu data XML file, categories and food items are displayed as shown in the figure above.

This screen allows the user to create, edit and delete different items and categories. It also allows the user to navigate back and forth to the main menu to save or load XML data files.

Button - Start Screen
This button allows the user to navigate back to the main screen.

Button - Add Category
This button allows the user to define a new category using the screen shown below.
Only the Name field for this new category must be entered. Optionally, a color can be selected for this category. This option brings up a prompt screen to select this color from a graphical palette, or enter its values in HSV, HSL, RGB, or CMYK formats. A layout template can also be selected, using a pull-down menu, if multiple templates have been defined in the layouts data file.

You may also select a default printer number for this category (Printer_1 or Printer_2) using a pull-down menu option.

Selecting the Done button prompts the user to save any changes and returns to the previous screen.
**Button - New Item**
This button allows the user to define a new item for the selected category.

All the fields defined for the item using either the “SupportedLabelFields” tag in the XML menu data file or the “MasterEditLayout” in the XML layout file are displayed to the user for data entry, in a tabular format.

![Item Data](image)

**Button - Previous Screen/Next Screen**
These buttons can be used to navigate back and forth between item screens if a large number of items are defined for a category.

**Category buttons**
Each of the categories defined in the menu data XML file is available as an independent button at the bottom of the screen. Selecting the category button allows the user to list the items defined for that category. This information is displayed in the upper part of the screen.
As shown in the figure above, a right click on the category button displays a menu that allows the user to either Delete Category, Edit Category, or Add New Field, the latter of which adds a field definition to the existing category definition. Clicking on any of the category buttons also allows the category selected to be edited by the user, as documented previously in this section.

**Food Item Buttons**

Food items defined for each category are displayed in the upper part of the Items/Categories screen. As the category selection changes, the items displayed are also updated based on the category selected.

Food items can be edited by either clicking on them, or by right-clicking them and selected the Edit Item function. A single click on the food item itself displays the food item edit screen shown below.
This screen shows the item data in a tabular format, allowing the user to modify food item values in the right-hand table column. Selecting the Done button navigates the user back to the Items/Categories screen, after prompting to saves any changes made by the user.
A right click on the food item displays a context menu, as shown in the figure above. This food item menu prompts for one of the following functions:

**Edit Item (List Screen)**
This option is same as a single left click on the food item itself and allows the user to edit the selected item using the tabular Item Data screen discussed earlier.

Selecting the Done button navigates the user back to the Items/Categories screen, and saves any changes made by the user.

After the user finishes modifying the data values for the food item, clicking the Done button navigates the user back to the Items/Categories screen after saving the user's changes.

**Duplicate Item**
This option allows the user to create a duplicate item belonging to the same category and with same data values as the original item. As shown in the figure below, this screen prompts the user to enter the duplicate item name. The Done button navigates back to the Items/Categories screen after creating the duplicate item.
Delete Item
This option deletes the selected food item, removing it from both the display and the menu data XML file.

Note: Items are deleted immediately when this function is used, with no prompting beforehand. If an item is deleted in error, consider exiting the application without saving the file, to revert to the previous configuration of the menu data XML file.
Appendix A: Sample XML Files

Sample XML File for Food Menu data

This file is a sample of a small food label input file. It defines field values per item of Caption, ShelfLife, Price, and Ingredients (shown in purple), categories of Breakfast, Salads, and Soups (shown in blue), and specific food items with assigned category, ingredients, and shelf life values (shown in pink).

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<LabelDataFile>
  <HeaderPC>
    <AppVersion>1.0.0.0</AppVersion>
  </HeaderPC>
  <HeaderPrinter>
    <CurrentFirmware>1.2.3</CurrentFirmware>
  </HeaderPrinter>
  <Shared>
    <SubsectionTag1>
      <Modify>false</Modify>
      <CreationDate>08-29-2012</CreationDate>
    </SubsectionTag1>
    <SubsectionTag2>
      <ModifiedDate>08-29-2012</ModifiedDate>
      <ModifiedBy>ShadowPrinterApplication</ModifiedBy>
    </SubsectionTag2>
    <SupportedLabelFields>
      <Caption/>
      <ShelfLife/>
      <Price/>
      <Ingredients/>
    </SupportedLabelFields>
  </Shared>
  <Categories>
    <AssignedCategory>
      <Name>Breakfast</Name>
      <Color>#ffff33</Color>
      <Language>English</Language>
      <PrintTemplate>Nutritional</PrintTemplate>
      <Printer>Printer_1</Printer>
    </AssignedCategory>
    <AssignedCategory>
      <Name>Salads</Name>
      <Color>#ffe800</Color>
      <Language>English</Language>
      <PrintTemplate>Nutritional</PrintTemplate>
      <Printer>Printer_1</Printer>
    </AssignedCategory>
    <AssignedCategory>
      <Name>Soups</Name>
      <Color>#330033</Color>
      <Language>English</Language>
      <PrintTemplate>Nutritional</PrintTemplate>
      <Printer>Printer_1</Printer>
    </AssignedCategory>
  </Categories>
</LabelDataFile>
```
<Language>English</Language>
(PrintTemplate>Nutritional</PrintTemplate>
(Printer>Printer_1</Printer>
</AssignedCategory>
</Categories>
</Labels>
<Item>
  <Caption>Bagels</Caption>
  <AssignedCategory>Breakfast</ AssignedCategory>
  <Ingredients>Wheat flour, sugar, butter</Ingredients>
  <ShelfLife>1 Day</ShelfLife>
</Item>
<Item>
  <Caption>Caesar Salad</Caption>
  <AssignedCategory>Salads</AssignedCategory>
  <Ingredients>Lettuce, tomatoes, onions</Ingredients>
  <ShelfLife>4 HRS</ShelfLife>
</Item>
<Item>
  <Caption>Minestrone soup</Caption>
  <AssignedCategory>Soups</AssignedCategory>
  <Ingredients>beans, tomatoes</Ingredients>
  <ShelfLife>2 DAYS</ShelfLife>
</Item>
<Item>
  <Caption>Mixed salad</Caption>
  <AssignedCategory>Salads</AssignedCategory>
  <Ingredients>Spinach, lettuce, arugula, onions</Ingredients>
  <ShelfLife>1 Day</ShelfLife>
</Item>
<Item>
  <Caption>Muffins</Caption>
  <AssignedCategory>Breakfast</AssignedCategory>
  <Ingredients>Flour, butter</Ingredients>
  <ShelfLife>2 Days</ShelfLife>
</Item>
<Item>
  <Caption>Vegetable Soup</Caption>
  <AssignedCategory>Soups</AssignedCategory>
  <Ingredients>Spinach, tomatoes, beans, carrots</Ingredients>
  <ShelfLife>2 DAYS</ShelfLife>
</Item>
</Labels>
</LabelDataFile>
Sample XML File for Food Label Layout

This XML file defines a single layout named "MasterEditLayout", and then static and dynamic text fields for the entries Description1, Description2, Ingredients, ShelfLife, and Price at specific label positions.

```xml
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<root>
  <Header></Header>
  <Layouts>
    <LayoutDefinition>
      <LayoutName>MasterEditLayout</LayoutName>
      <LayoutWidth>30</LayoutWidth>
      <LayoutHeight>30</LayoutHeight>
      <LayoutRotation>0</LayoutRotation>
      <StaticText>
        <RectXPos>5</RectXPos>
        <RectYPos>2</RectYPos>
        <RectLen>70</RectLen>
        <RectHeight>7</RectHeight>
        <RectCrop>TRUE</RectCrop>
        <FontName>Arial</FontName>
        <FontSize>18</FontSize>
        <FontFlags>TEXT_BOLD</FontFlags>
        <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
        <AlignmentFlags>ALIGN_CENTER</AlignmentFlags>
        <Text>Description1</Text>
      </StaticText>
      <StaticText>
        <RectXPos>5</RectXPos>
        <RectYPos>12</RectYPos>
        <RectLen>70</RectLen>
        <RectHeight>7</RectHeight>
        <RectCrop>TRUE</RectCrop>
        <FontName>Arial</FontName>
        <FontSize>18</FontSize>
        <FontFlags>TEXT_BOLD</FontFlags>
        <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
        <AlignmentFlags>ALIGN_CENTER</AlignmentFlags>
      </StaticText>
      <DynamicText>
        <RectXPos>38</RectXPos>
        <RectYPos>2</RectYPos>
        <RectLen>70</RectLen>
        <RectHeight>7</RectHeight>
        <RectCrop>TRUE</RectCrop>
        <FontName>Arial</FontName>
        <FontSize>14</FontSize>
        <FontFlags>TEXT_BOLD</FontFlags>
        <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
        <Name>Description1</Name>
      </DynamicText>
    </LayoutDefinition>
  </Layouts>
</root>
```
<Text>Description2</Text>
</StaticText>

<DynamicText>
  <RectXPos>38</RectXPos>
  <RectYPos>12</RectYPos>
  <RectLen>70</RectLen>
  <RectHeight>7</RectHeight>
  <RectCrop>TRUE</RectCrop>
  <FontName>Arial</FontName>
  <FontSize>14</FontSize>
  <FontFlags>TEXT_BOLD</FontFlags>
  <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
  <Name>Description2</Name>
</DynamicText>

<StaticText>
  <RectXPos>5</RectXPos>
  <RectYPos>22</RectYPos>
  <RectLen>70</RectLen>
  <RectHeight>7</RectHeight>
  <RectCrop>TRUE</RectCrop>
  <FontName>Arial</FontName>
  <FontSize>18</FontSize>
  <FontFlags>TEXT_BOLD</FontFlags>
  <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
  <AlignmentFlags>ALIGN_CENTER</AlignmentFlags>
  <Text>Ingredients</Text>
</StaticText>

<DynamicText>
  <RectXPos>38</RectXPos>
  <RectYPos>22</RectYPos>
  <RectLen>70</RectLen>
  <RectHeight>7</RectHeight>
  <RectCrop>TRUE</RectCrop>
  <FontName>Arial</FontName>
  <FontSize>14</FontSize>
  <FontFlags>TEXT_BOLD</FontFlags>
  <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
  <Name>Ingredients</Name>
</DynamicText>

<StaticText>
  <RectXPos>5</RectXPos>
  <RectYPos>32</RectYPos>
  <RectLen>70</RectLen>
  <RectHeight>7</RectHeight>
  <RectCrop>TRUE</RectCrop>
  <FontName>Arial</FontName>
  <FontSize>18</FontSize>
  <FontFlags>TEXT_BOLD</FontFlags>
  <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
  <AlignmentFlags>ALIGN_CENTER</AlignmentFlags>
  <Text>ShelfLife</Text>
</StaticText>
Appendix A: Ordering Supplies

<DynamicText>
  <RectXPos>38</RectXPos>
  <RectYPos>32</RectYPos>
  <RectLen>70</RectLen>
  <RectHeight>7</RectHeight>
  <RectCrop>TRUE</RectCrop>
  <FontName>Arial</FontName>
  <FontSize>14</FontSize>
  <FontFlags>TEXT_BOLD</FontFlags>
  <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
  <Name>ShelfLife</Name>
</DynamicText>

<StaticText>
  <RectXPos>5</RectXPos>
  <RectYPos>42</RectYPos>
  <RectLen>70</RectLen>
  <RectHeight>7</RectHeight>
  <RectCrop>TRUE</RectCrop>
  <FontName>Arial</FontName>
  <FontSize>18</FontSize>
  <FontFlags>TEXT_BOLD</FontFlags>
  <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
  <AlignmentFlags>ALIGN_CENTER</AlignmentFlags>
  <Text>Price</Text>
</StaticText>

<DynamicText>
  <RectXPos>38</RectXPos>
  <RectYPos>42</RectYPos>
  <RectLen>70</RectLen>
  <RectHeight>7</RectHeight>
  <RectCrop>TRUE</RectCrop>
  <FontName>Arial</FontName>
  <FontSize>14</FontSize>
  <FontFlags>TEXT_BOLD</FontFlags>
  <FontStyle>STYLE_SERIF:STYLE_EMPHASIS</FontStyle>
  <Name>Price</Name>
</DynamicText>

</LayoutDefinition>
</Layouts>
</root>
## Index

<table>
<thead>
<tr>
<th>About the menu editor, 4</th>
<th>Main screen functions, 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copyright, iii</td>
<td>Overview of application, 7</td>
</tr>
<tr>
<td>Data files, overview, 11</td>
<td>Running the application, 7</td>
</tr>
<tr>
<td>Data, overview, 9</td>
<td>Sample XML files, 21</td>
</tr>
<tr>
<td>Disclaimer, iii</td>
<td>Menu data file, 21</td>
</tr>
<tr>
<td>Items/Categories screen functions, 14</td>
<td>Layout data file, 23</td>
</tr>
<tr>
<td></td>
<td>Trademarks, iii</td>
</tr>
</tbody>
</table>
Index