

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 July 2002 (11.07.2002)

PCT

(10) International Publication Number
WO 02/054653 A2

(51) International Patent Classification⁷:

H04L

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(21) International Application Number:

PCT/US02/00253

(22) International Filing Date:

7 January 2002 (07.01.2002)

(25) Filing Language:

English

(26) Publication Language:

English

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(30) Priority Data:

60/259,867 8 January 2001 (08.01.2001) US

(71) Applicant: **BETHLEHEM STEEL CORPORATION**

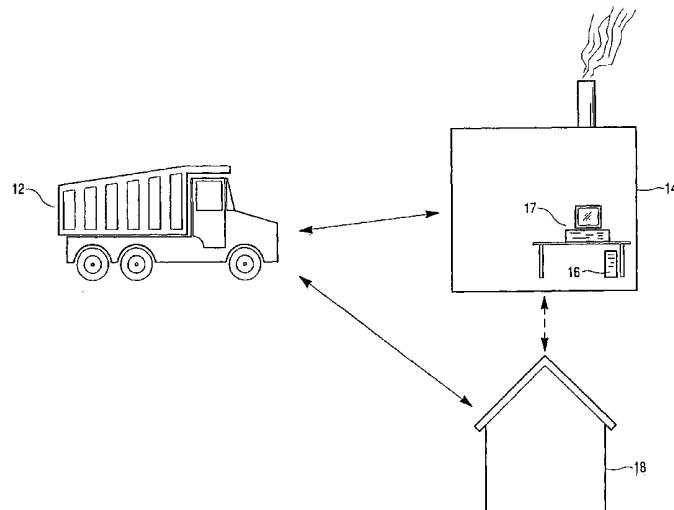
[US/US]; 1170 Eighth Avenue, Room 1969 Martin Tower, Bethlehem, PA 18016-7699 (US).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: CREDIT INQUIRY METHOD AND SYSTEM



WO 02/054653 A2

(57) **Abstract:** An automated system for disseminating information from a company to a third party about customers of the company is disclosed. The information disseminated relates to a relationship between the company and one of its customers. The company provides a website having a secure portion that may be accessed only via a previously assigned ID and password. The company provides the Customer Account Number, but the Commercial Identification Number may be assigned or changed by the customer. The customer thus controls access to the information to be provided to a third party, because it must supply both numbers to those who are to be able to access the stored information. The system eliminates the necessity of manually completing requests. By automating the system information retrieval is consistent, quicker, and allows for central administration control and monitoring of all inquiries. Response to third parties wishing to obtain information is almost instantaneous.

CREDIT INQUIRY METHOD AND SYSTEM

5

CROSS REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/259,867, filed January 8, 2001, the disclosure of which is incorporated herein by reference.

10

REFERENCE TO COMPUTER PROGRAM LISTING APPENDIX

The computer program listing appendix container in the file "Sourcecode.txt" on compact disc 1 of 1, which has been filed with the United States Patent and Trademark Office in duplicate, is hereby incorporated by reference. The file was created on December 17, 2001, and is 263,153 bytes in size.

15

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to the secure dissemination of information between entities. More specifically, this invention is directed to implementation of a website by one corporation in order to permit secure access to credit information of its customers by third parties authorized by the customers to have access to that credit information.

25

2. Description of Related Art

Often times a first company will contact a second company to inquire about a customer's business relationship with that second company. The inquiry seeks information about the credit history between the customer and the second company. The first company may request a report about the about the credit history between the second company and the customer. The second company will provide the first company with some or all the requested information or like information. The second company usually is not obligated to provide the credit information, but does so as a service to its own customers. This service is beneficial to the second company's customers, who seek credit with entities requiring credit references before extending

any form of credit or business.

- Contact is usually made by telephone, facsimile, letter, or e-mail. In order to respond to an inquiry, the second company must allocate resources in the form of manpower, time, and money to answer the individual requests. Generally, the larger 5 the company, the greater the volume of requests it receives. Supplying the information, while a benefit for customers, creates a burden on the second company's staff, consuming up to twenty percent of an employee's available work time. Employees must retrieve the information from a computer or other source and communicate the information to the first company via phone, letter, e-mail or 10 facsimile. Additionally, if the second company has more than one plant or office, all relevant information may not be centrally located, and additional time and resources will be required in order to provide the requested information. Information that is held at various locations may not be included in the response due to its unavailability or lack of knowledge on the part of the individual compiling the information. 15 Moreover, where the second company has multiple locations, each location might respond differently to a request, possibly creating confusion and inconsistency. Also, the individual charged with providing the information might not be available, possibly delaying the response and causing the customer to be denied credit or business.

In addition to increasing overhead for the second company, sometimes the 20 request does not come from a legitimate source. A company may request credit information for purposes other than those communicated. For example, a company may wish to determine why a customer has failed to meet the normal volume for a given period. The current methods of providing credit information do not prevent a competitor from seeking a credit report from another company about its business with 25 a particular customer. Companies in competition often seek the type of information provided. A customer's relationship with other companies may be affected by this freely obtained information. In order to prevent responding to such improper inquiries, a company would have to spend an inordinate amount of time and resources checking on whether each inquiry was legitimate.

30

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method wherein predetermined information about a customer is stored on a storage device under the control of a host. Access to the predetermined information is limited by creating a

general identification and a specific identification specific to the customer. The general identification links the predetermined information to the customer. The specific identification is used by the customer to control the access to the predetermined information. A customer informs a third party of the general and 5 specific identification, thereby allowing the third party to gain access to the predetermined information.

The present invention also provides a computer system allowing access to predetermined information. The computer system stores the predetermined information on a storage device of a host's computer. Within the computer system is 10 a general and a specific identification particular to a company, e.g., a customer. The general identification links the predetermined information to the customer. The specific identification allows for access to the predetermined information. A third party accesses the computer system to view the predetermined information, after being provided the general and specific identification by the customer.

15 The present invention also pertains to secure and selective computer access to credit information between a first company and a second company with which the first company does business. The credit information is stored on a storage device, and access to the credit or other information is limited. A general identification is created for the first company when it begins doing business with the second company, and 20 credit history and other information about the first company is associated with this general identification. The first company also creates a specific identification used to control access to the information associated with its general identification. The general and specific identifications also limit access to the credit information. A third party may access the credit information on the storage device only after providing the 25 general and specific information.

An advantage of the present invention is nearly complete automation of the inquiry process, thus significantly reducing the need to manually complete requests. In applicant's experience, the number of credit inquiries that had to be handled manually was reduced by over 75% once the system was implemented. By 30 automating the process, information retrieval is consistent, quicker, and allows for central administration control and monitoring of all inquiries. In addition, a log can be kept so that the various incoming inquiries can be reviewed as needed. Response to third parties wishing to obtain information is almost instantaneous. Costs associated with maintaining this service are substantially less than the costs associated

with the prior manual response system.

These advantages and other novel features of the present invention will become apparent in the following detailed description of the invention when considered in conjunction with the accompanying drawings.

5

BRIEF DESCRIPTIONS OF THE DRAWINGS

A better understanding of the present invention will be had when reference is made to the accompanying drawings, wherein identical parts are identified by identical reference numbers and wherein:

10 FIG. 1 is schematic diagram depicting the relationship between several entities;

FIG. 2 is a flow chart setting forth the process steps of the invention; and

FIG. 3 is a flow chart setting forth the process for accessing a host's website.

15

DETAILED DESCRIPTION OF THE INVENTION

The features of the invention as explained above provide an innovative technique for providing information about a business relationship between two entities to a third party. As best shown in Figure 1, a customer 12 conducts business 20 with a host 14. The customer 12 may wish to use the established relationship with the host 14 as a reference. The customer 12 may wish a third party 18 to be provided with information relating to the relationship, e.g., credit history, between the customer 12 and the host 14.

25 The host 14 maintains information concerning the transactions occurring during the business relationship with customer 12. Preferably the information is stored within a storage device 16 on the host's computer 17. The storage device may include computer RAM, a hard drive, a disk, an independent server, etc. Computer storage of the information permits access by numerous individuals or entities, and permits the host 14 to control access to the information, the format of the information, 30 and the precise information to be made available.

The stored information may relate to the credit history between the customer 12 and the host 14. The credit history includes information such as largest credit amount extended to the customer, length of time required for the customer 12 to pay debts, the duration during which the customer 12 and host 14 have been conducting

transactions, general terms of sale, and the date of the last transaction between the customer 12 and the host 14. Other information, of course, could also be stored. All or selected portions of this information may be made available to third parties. For example, while the information may include data on all transactions between the 5 customer and the host, only data from the past five years, or data concerning certain types of transactions, for example, may be made available to third parties.

As best shown in Figure 2, the host 14 assigns a general identification number to customer 12 at 20 when the customer opens an account or begins doing business with the host. Information about the customer's relationship with the host, credit 10 information, for example, is thereafter stored in a file associated with this general identification number at step 30. This general identification number is known as the Customer Account Number ("CAN"), and only the host 14 may alter or change the customer's assigned CAN. The CAN may be a sequence of any length comprised of any combination of numbers and/or letters. The CAN preferably comprises no more 15 than ten alphanumeric characters. Information about customer 12 stored on computer 16 is linked to that customer 12 by the CAN.

Preferably host 14 maintains a website that may be accessed via the Internet or other communications system. Customers may access a secure area within the main website. Alternatively, the host 14 may establish a dial-up connection for customers 20 via standard phone connections. Each customer 12 has selective access to the secure area established by the host 14, via a customer ID and password assigned at step 40.

In addition to the CAN created at 20, a specific identification is created by or for customers desiring to use host 14 as a credit reference. The specific identification is known as the Commercial Identification Number ("CIN"). The CIN may be 25 comprised of numbers and/or letters. The CIN is limited preferably to no more than ten alphanumeric characters. Only the customer 12 and the host 14 initially know the CIN. There are several ways in which a customer 12 may obtain a CIN. The host 14 may assign a CIN to a customer 12, the customer 12 may assign its own CIN within the secure area of host 14, or the customer 12 may request that the host 14 assign a 30 particular CIN. The host 14 then confirms to customer 12 the CIN that has been assigned. The customer 12 sets its own CIN or may receive the CIN from the host 14 via mail, e-mail, facsimile, telephone, in person, etc. Once the CAN and CIN have been established, then the website of the host 14 may be accessed by the customer and/or third parties to whom the customer provides the CAN and CIN. The

information that may be accessed is limited by the host 14, and access to the secure area of the website does not permit unrestricted access to the entirety of the website or to the computer network of the host 14. The CIN may be changed by the customer 12, thus permitting the customer to control access by third parties to its information.

- 5 The customer 12 accesses the secure portion of the website of host 14 commencing at step 50. The customer is prompted to enter a previously applied-for and assigned ID and password at 52 and access is denied at step 54 if the ID or password is incorrect. Within the secure area at 60 there are instructions and ability for the customer 12 to initially choose and set a CIN, change the CIN, or delete the
10 CIN permanently. This allows the customer 12 to control the access to the stored information on the website by third parties 18.

Rather than having the third party 18 submit requests to the host 14 regarding the information retained by host 14 about customer 12, the third party 18 can access the website of the host 14 in order to obtain the information about the customer 12.

- 15 The customer 12 must inform the third party 18 of the correct CAN and CIN in order to allow the third party 18 to gain access to the information stored on the host's computer through the website. In this way, the host 14 does need to dedicate personnel to responding to inquiries and can control the types of information that it provides. Additionally, the customer 12 controls who can access the information,
20 thus substantially eliminating the ability of a competitor of host 14 to inquire about customer 12. Also, because access is achieved by Internet communications, information can be obtained at a time convenient to the third party.

- As best shown in Figure 3, the third party 18 attempts to access the host's website at 150. The third party 18 is prompted to enter the CAN and CIN that correspond to the customer 12 at step 152. If either CAN or CIN, or both, are incorrect then access is denied at step 154. If both CAN and CIN are correct, the third party 18 can access the information pertaining to the customer 12 stored by the host 14. The computer 17 of the host 12 automatically retrieves the stored or predetermined information from the daily updated records, and provides this
30 information to third party 18 at step 160.

When a third party 18 attempts to obtain access to the stored information via the website, the host 14 may request that certain information be provided prior to access being granted. This would be requested at or prior to step 152. Such information may include the third party's name and address, Federal identification

number, Data Universal Numbering System (DUNS) number, the name and title of the individual making the request, and any other information host 14 wants to obtain. This information, as well as the date of the inquiries and number of inquiries made is stored in a log that can be accessed as needed.

5 The customer 12 may limit or terminate access by a third party 18 to its information by changing its CIN. Customer 12 may, for example, change its CIN on a periodic basis. Alternatively, it may change it within a defined period after having supplied it to a third party 18. Access of third parties 18 through changing the CIN allows customer 12 to control who can access its information.

10 Figure 1 illustrates the flow of information between the host 14, customer 12 and the third party 18. Information flow requiring human interaction or involvement is indicated by a line, with arrows indicating the direction of information flow. The host 14 and the customer 12 communicate information for every business transaction. The customer 12 and third party 18 similarly interact during business relationships.

15 By automation of the above-mentioned process, the host 14 communicates with the third party 18 only through electronic means, as indicated by the dashed lines, saving valuable resources.

20 While the foregoing invention has been shown and described with reference to several preferred embodiments, it will be understood that various changes in form and detail may be made without departing from the spirit and scope of the present invention.

Subject	Body	From: (Name)	From: (Address)	From: (Type)
To: (Name)	To: (Address)	To: (Type)	CC: (Name)	CC:
(Address)	CC: (Type)	BCC: (Name)	BCC: (Address)	BCC: (Type)
	Billing Information	Categories	Importance	Mileage
	Sensitivity			
CreditPatent	"Table of Contents			
I. Class Source Files				
.....4				
II. HTML Source Files				
.....4				
i.	AppProcessPg.htm			4
ii.	AppResults.htm			4
iii.	CreditApp2.htm			5
iv.	CreditApp3.htm			8
v.	CreditApp4.htm			11
vi.	CreditApplication.htm			12
vii.	CreditDeptHomePg.htm			15
viii.	CreditDiscountPolicy.htm			16
ix.	CreditInquiry.htm			17
x.	CreditInquiryForm.htm			18
xi.	CreditInquiryResults.htm			21
xii.	CreditPolicy.htm			22
xiii.	CreditRepNewCust.htm			23
xiv.	CreditRepNewResults.htm			24
xv.	InquiryTrackScreen.htm			25
xvi.	Override.htm			25
xvii.	pgCreditInstruct.htm			26
xviii.	pgError.htm			28
III. JAVA Source Files				
i.	AppProcessPg.java			28
ii.	AppResults.java			29
iii.	CBethIMSDataField.java			44
iv.	CBethIMSFiller.java			48
v.	CBethIMSPadRight.java			49
vi.	CBethIMSPprocedure.java			49
vii.	Credit.java			60
viii.	CreditApp.java			89

ix.	CreditApp2.java	89
x.	CreditApp3.java	97
xi.	CreditApp4.java	103
xii.	CreditApplication.java	110
xiii.	CreditDeptHomePg.java	114
xiv.	CreditDiscountPolicy.java	115
xv.	CreditInquiry.java	116
xvi.	CreditInquiryForm.java	116
xvii.	CreditInquiryResults.java	125
xviii.	CreditPolicy.java	126
xix.	CreditRepNewCust.java	127
xx.	CreditRepNewResults.java	130
xxi.	EmailObj.java	131
xxii.	imsCE4ZWB67.java	132
xxiii.	imsCE4ZWB68.java	133
xxiv.	imsDataSource.java	133
xxv.	imsUtilitea.java	134
xxvi.	InquityTrackScreen.java	134
xxvii.	MsSqlDataSource.java	135
xxviii.	Override.java	135
xxix.	PayDAyObj.java	140
xxx.	pgCreditInstruct.java	140
xxxi.	pgError.java	141
xxxii.	sqlCINReference.java	144
xxxiii.	sqlCustCode.java	144
xxxiv.	sqlDistrict.java	144
xxxv.	sqlInquiryOverride.java	145
xxxvi.	sqlInquiryTracking.Java	145
IV.	Metadata Source File	
i.	IMSService.metadata	145
V.	Project Files	
		195
VI.	SDO Source Files	
i.	imsCE4ZWB67.sdo	190
ii.	imsCE4ZWB68.sdo	

.....	194
iii. sqlCINReference.sdo	198
iv. sqlCustCode.sdo	200
v. sqlDistrict.sdo	201
vi. sqlInquiryOverride.sdo	202
vii. sqlInquiryTracking.sdo	204
VII. SDS Source Files	
i. ImsDataSouces.sds	210
ii. MsSqlDataSource.sds	211
VIII. SER Source Files	
.....	212
IX. SPG Source Files	
i. AppProcessPg.spg	212
ii. AppResults.spg	215
iii. CreditApp2.spg	233
iv.	
CreditApp3.spg	251
v.	
CreditApp4.spg	262
vi. CreditApplication.spg	270
vi. CreditDeptHomePg.spg	276
viii. CreditDiscountPolicy.spg	279
ix. CreditInquiry.spg	280
x. CreditInquiryForm.spg	281
xi. CreditInquiryResults.spg	290
xii. CreditPolicy.spg	294
xiii. CreditRepNewCust.spg	295
xiv. CreditRepNewResults.spg	297
xv. InquiryTYrackSrceen.spg	300
xvi. Override.spg	306
xvii. pgCreditInstruct.spg	309
xviii. pgError.spg	310
X. Sum Source File	
i. CBethIMSDataField.sum	312
ii. CBethIMSFiller.sum	312
iii. CBethIMSPADRight.sum	312
iv. CBethIMSPprocedure.sum	

.....	312
v. Credit.sum 312
vi. EmailObj.sum 313
vii. PayDayObj.sum 313
XI. SID Source File
.....	314

I. Class Source Files - These files will be constructed at compile time based on the classes specified in the application.

II. HTML Source Files

i.

```
AppProcessPg.htm
<HTML>
<HEAD>
<TITLE>AppProcessPg</TITLE>
</HEAD>
[[SPIDERPAGE NAME=AppProcessPg]]
<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=""10"" align=center>**logobeth**
</td></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center><font size=4 color=""ff0000"">Your
application was sent successfully. Thank you</td></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>**GoToCreditHmPg**</td></tr>
</TABLE>
**SPIDERSESSION**
[[/SPIDERPAGE]]
</BODY>
</HTML>
```

ii.

```
AppResults.htm
<HTML>
<HEAD>
<TITLE>Application Results</TITLE>
</HEAD>
[[SPIDERPAGE NAME=AppResults]]
```

```

<center>

<TABLE BORDER=""0"" CELLSPACING=""0"" CELLPADDING=""0"" VALIGN=TOP
WIDTH=650>
<tr>
<td colspan=""10"" width=650 align=left><font size=4 color=""Red"">
<B>You did not fill in these required fields:</B></td></tr>
<tr>
<td colspan=""10"" width=650 align=left>&nbsp;</td></tr>
<tr>
<th colspan=""10"" width=650 align=left><font size=3 color=""black"">
**CpnyNameLabel**</font></th></tr>

<tr>
<td colspan=""10"" width=650 align=left>
**IncorpName****IncorporateName**</td></tr>
<br>

<th colspan=""10"" width=650 align=left><font size=3 color=""black"">
**HeadAddLabel**</font></th></tr>
<tr>
<td colspan=""10"" width=650 align=left>**HeadStLabel****HeadStAdd**
</td></tr>
<tr>
<td colspan=""10"" width=650 align=left>
**HeadCityLabel****HeadquCity**&nbsp;&nbsp;
**HeadquStLabel****HeadState**&nbsp;&nbsp;
**HeadquZipLabel****HeadZip**</td></tr>
<tr>
<td colspan=""10"" width=650 align=left>**HeadPhLabel****HeadquPh**
</td>
</tr>

<br>

<tr>
<th colspan=""10"" width=650 align=left><font size=3 color=""black"">
**DunFedLabel**</font></th></tr>
<tr>
<td colspan=""10"" width=650 align=left>**DunsLabel****Duns1****Duns2
****Duns3**</td></tr>
<tr>
<td colspan=""10"" width=650 align=left>**FedLabel****Fed1**&nbsp;
**Fed2**</td></tr>

<br>

<tr>
<th colspan=""10"" width=650 align=left><font size=3 color=""black"">
**BankRefLabel**</font></th></tr>
<tr><td colspan=""10"" width=650 align=left>**BankNameLb****BankName1
**</td></tr>
<tr><td colspan=""10"" width=650 align=left>
**BankStLabel****BankStr**</td></tr>
<tr><td colspan=""10"" width=650 align=left>**BankCityLb****BankCity1
**&nbsp;**BankStLb****BankSt1**&nbsp;**BankZipLb****BankZip1**</td>
</tr>
<br>
<br>
<tr>
<th colspan=""10"" width=650 align=left><font size=3 color=""black"">
**TradeRefLabel**</font></th></tr>
<tr>
<td colspan=3 align=left>**Ref1Lab****Ref1NameLabel**</td>
<td colspan=7 align=left>**RefName1**</td></tr>
<tr>

```

```
<td colspan=3 align=left>&ampnbsp**Ref1CityLabel**</td>
<td colspan=7 align=left>**RefCity1**</td></tr>
<br>
<tr>
<td colspan=3 align=left>**Ref2Lab****Ref2NameLabel**</td>
<td colspan=7 align=left>**RefName2**</td></tr>
<br>
<tr>
<td colspan=3 align=left>&ampnbsp**Ref2CityLabel**</td>
<td colspan=7 align=left>**RefCity2**</td></tr>
<br>
<tr>
<td colspan=3 align=left>**Ref3Lab****Ref3NameLabel**</td>
<td colspan=7 align=left>**RefName3**</td></tr>
<br>
<td colspan=3 align=left>&ampnbsp**Ref3CityLabel**</td>
<td colspan=7 align=left>**RefCity3**</td></tr>
<br>
<tr><td colspan=10 align=center>**Submit**</td></tr>
</table>
<BR>**From**
<BR>
<BR>
**SPIDERSESSION**
[[/SPIDERPAGE]]
</BODY>
</HTML>
```

iii.

```
<tr>
<td colspan=""10"" align=left>Street Address <b>Required</b>&nbsp;
&nbsp;**HeadquarterAddress**</td></tr>
                                |
<tr>
<td colspan=""10"" align=left>Box #
&nbsp;&nbsp;&nbsp;**HeadquarterBoxNum**</td></tr>

<tr>
<td colspan=""10"" align=left>City, State, Zip&nbsp;<b>Required</b>
**HeadquarterCity**&nbsp;**HeadquarterState**&nbsp;**HeadquarterZip**
</td></tr>

<tr>
<td colspan=""10"" align=left>Foreign if applicable
&nbsp;&nbsp;&nbsp;**HeadquarterForeignAddress**</td></tr>

<tr>
<td colspan=""10"" align=left>Phone &nbsp;<b>Required &nbsp;</b>
**HeadquarterPhone**&nbsp;&nbsp;&nbsp;&nbsp;
Fax&nbsp;**HeadquarterFax**</td></tr>

<tr>
<td colspan=""10"" align=left>Web Address/ www.*HeadquarterWeb*
</td></tr>

<tr>
<td colspan=""10"">&nbsp;</td></tr>

<tr>
<th colspan=""10"" align=left bgcolor="#666699"><font size=4
color="white">&nbsp;Branch Location</font></th></tr>

<tr>
<td colspan=""10"" align=left>Are you planning to purchase through a
branch location?
&nbsp;&nbsp;&nbsp;**PurchaseThroughBranch**</td></tr>

<tr>
<td colspan=""10"" align=left>&nbsp;</td></tr>

<tr>
<td colspan=""10"" align=left>If yes, please enter the Branch name
and address:</td></tr>

<tr>
<td colspan=""10"" align=left>Name&nbsp;&nbsp;&nbsp;**BranchName**
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Division&nbsp;&nbsp;&nbsp;&nbsp;
**BranchDivision**</td></tr>

<tr>
<td colspan=""10"" align=left>Street Address&nbsp;&nbsp;&nbsp;
**BranchStAddress**</td></tr>

<tr>
<td colspan=""10"" align=left>City, State, Zip&nbsp;&nbsp;&nbsp;
**BranchCity**&nbsp;
&nbsp;**BranchState**&nbsp;&nbsp;**BranchZip**</td></tr>

<tr>
<td colspan=""10"" align=left>Foreign Address if Applicable:&nbsp;
&nbsp;&nbsp;**BranchForeignAddress**</td></tr>

<tr>
<td colspan=""10"" align=left>Phone&nbsp;&nbsp;&nbsp;**BranchPhone**
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Fax&nbsp;&nbsp;&nbsp;**BranchFax**
```



```
**SPIDERSESSION**
[[/SPIDERPAGE]]

</BODY>
</HTML>                                         iv.

CreditApp3.htm
<HTML>
<HEAD>
<TITLE>Credit Application</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditApp3]]

<center>

<TABLE BORDER=""0"" CELLSPACING=""0"" CELLPADDING=""0"" WIDTH=650>
<tr>
<th colspan=""10"" width=650 align=left bgcolor=""#666699""><font
size=4 color=""white"">&nbsp;Accounts Payable Information</font></th>
</tr>

<tr>
<td colspan=""10"" width=650 align=left>Is your Accounts Payable
system:
&nbsp;&nbsp;&nbsp;**ApSystem**</td></tr>

<tr>
<td colspan=""10"" width=650 align=left>&nbsp;</td></tr>

<tr>
<td colspan=""10"" width=650 align=left>Are payments mailed from:
&nbsp;&nbsp;&nbsp;**ApMailedTo**</td></tr>

<tr>
<td colspan=""10"" width=650 align=right>&nbsp;</td></tr>

<tr>
<td colspan=""10"" align=left width=650>Buyer's name (person)
&nbsp;**BuyersName**</td></tr>

<tr>
<td colspan=""10"" align=left width=650>&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Phone Number
&nbsp;**BuyersPhoneNum**</td></tr>
<tr>
<td colspan=""10"" align=left width=650>&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Email Address&nbsp;
**BuyersEmail**</td></tr>

<tr>
<td colspan=""10"" width=650 align=left>&nbsp;</td></tr>

<tr>
<td colspan=""10"" align=left width=650>Accounts Payable Supervisory
person's name
&nbsp;**ApSupervisorName**</td></tr>

<tr>
```



```
<tr><td colspan=""10"" align=left>City  
&nbsp;<b>-Required</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**BankCity**</td></tr>  
  
<tr>  
<td colspan=""10"" align=left>State  
&nbsp;<b>-Required</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
**BankState**  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Other&nbsp;&nbsp;**BankOther**</td></tr>  
<tr>  
<td colspan=""10"" align=left>Zip&nbsp;<b>-Required</b>&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
**BankZip**</td></tr>  
<tr>  
<td colspan=""10"" align=left>Account officer's name&nbsp;&nbsp;  
**AcctOfficerName**</td></tr>  
  
<tr>  
<td colspan=""10"" align=left>Phone Number&nbsp;&nbsp;  
**AcctOfficerPhoneNum**</td></tr>  
  
<tr>  
<td colspan=""10"" align=left>&nbsp;</td></tr>  
<tr>  
<td colspan=10>&nbsp;</td></tr>  
  
<tr>  
<td colspan=""10"" align=left><b>Note: </b><i>In today's business  
environment, banks are sometimes reluctant to respond  
to a prospective creditor's request, whether due to individual bank  
policy or time/manpower  
constraints. This significantly hampers our ability to complete a  
credit check. To expedite this  
process, we ask that you contact your bank and request them to fax  
the bank's experience/relationship  
to Bethlehem Steel Corporation at (610) 694-3287 or email to:  
**EmailBankInfo**</td></tr>  
</table>  
</td>  
  
<table border=0 cellpadding=2 cellspacing=2 width=650>  
<tr>  
<td colspan=10>&nbsp;</td></tr>  
<tr>  
<td colspan=10 align=right>**GoToApp2** &nbsp; | &nbsp;**GoToApp4**  
</td></tr>  
<tr>  
<td colspan=10 align=left>**GoToCreditHmPg**</td></tr></table>  
  
**SPIDERSESSION**  
[[/SPIDERPAGE]]  
  
</BODY>  
</HTML>
```

v.

```
CreditApp4.htm
<HTML>
<HEAD>
<TITLE>Credit Application</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditApp4]]

<center>
<table border=0 cellspacing=0 cellpadding=0 width=650>
<tr>
<th colspan=10 align=left bgcolor="#666699"><font size=4
color="white">&nbsp;Trade References</font></th></tr>
<tr>
<td colspan=10 align=left>Preferably other large volume suppliers.
&nbsp;<b>-Name and City/St/Zip are Required</b></td></tr>

<tr>
<td colspan=2 align=right>(1)&nbsp;&nbsp;Name:</td>
<td colspan=8 align=left>**TradeRefName1**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;Street:</td>
<td colspan=8 align=left>**TradeRefStreet1**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;City/St/Zip:</td>
<td colspan=8 align=left>**TradeRefCityStZip1**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Attn:<**TradeRefAttn1**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Phone:<**TradeRefPhone1**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Fax:<**TradeRefFax1**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=right>(2)&nbsp;&nbsp;Name:</td>
<td colspan=8 align=left>**TradeRefName2**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;Street:</td>
<td colspan=8 align=left>**TradeRefStreet2**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;City/St/Zip:</td>
<td colspan=8 align=left>**TradeRefCityStZip2**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Attn:<**TradeRefAttn2**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Phone:<**TradeRefPhone2**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Fax:<**TradeRefFax2**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=right>(3)&nbsp;&nbsp;Name:</td>
<td colspan=8 align=left>**TradeRefName3**</td></tr>
<tr>
```

```

<td colspan=2 align=right>&nbsp;&nbsp;Street</td>
<td colspan=8 align=left>**TradeRefStreet3**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;City/St/Zip:</td>
<td colspan=8 align=left>**TradeRefCityStZip3**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Attn:**TradeRefAttn3**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Phone:**TradeRefPhone3**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Fax:**TradeRefFax3**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10><hr color=""black"" width=650></td></tr>

<tr>
<td colspan=10 align=left><b>Thank you for your time in completing
this application. Click
the ""Submit"" button below and we will get to work on processing
your application as quickly
as possible. We look forward to doing business with you!</b></td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=center>**Submit**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;</td>
</td></tr></table>

<table border=0 cellpadding=2 cellspacing=2 width=650>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=right>**GoToApp3**</td></tr>
<tr>
<td colspan=10 align=left>**GoToCreditHmPg**</td></tr>
</TABLE>

**From**
**SPIDERSESSION**
[[/SPIDERPAGE]]

</BODY>
</HTML>

```

vi.

```

CreditApplication.htm
<HTML>
<HEAD>
<TITLE>Credit Application</TITLE>
</HEAD>
[[SPIDERPAGE NAME=CreditApplication]]
<center>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>

```

```

<td colspan=""10"" align=center>**logobeth**
</td></tr>
</TABLE>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<th colspan=""10"" align=center bgcolor="#666699"><font size=4
color="white">Credit Application</font></th>
</tr>

<tr>
<td colspan=""10"" align=center>&ampnbsp</td></tr>
<td colspan=""10"" align=center><i>(Note: We retain all information
in strict confidence)</i></td></tr>

<tr>
<td colspan=""10"" align=center>&ampnbsp</td></tr>
<td colspan=""10"" align=left>We want to sell to you! We do not want
"Credit" to be a reason to delay,
refuse, or lose your order. By filling out all areas of our on-line
credit application, we will be
able to handle the credit process in the fastest possible manner. We
encourage electronic submission,
however, you may use the print option and fax the completed form to:
(610) 694-3287</td></tr>

<tr>
<td colspan=""10""><hr width=""650"" color="#666699"></td></tr>
<tr>
<th colspan=""10"" align=left><b>Note: If your initial purchase is
intended to be via <i>Metalsite</i>, please
fill out the credit application form through the 'Buyer
Registration' process at **MetalSiteLink**</td></tr>
<tr>
<td colspan=""10""><hr width=""650"" color="#666699"></td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=""650">
<tr>
<td colspan=""10"">&ampnbsp</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>Have you ever purchased
from Bethlehem in the past?
&ampnbsp&ampnbsp&ampnbsp&ampnbsp**PastCust**</td></tr>

<td colspan=""10"">&ampnbsp</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>If yes, when and under
what name?
&ampnbsp&ampnbsp&ampnbspYear- **PastYear**&ampnbsp&ampnbsp&nbsp;Name- **PastName**</td></tr>

<tr>
<td colspan=""10"" width=""650"">&ampnbsp</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>What product or products
do you plan to purchase initially?</td></tr>

<tr>
<td colspan=""2"" width=""190"" align=left>**Plate**Plate</td>
<td colspan=""2"" width=""200"" align=left>
**SemiFinished**SemiFinished</td>
<td colspan=""2"" width=""190"" align=left>**Pipe**Pipe</td>

```

```

<td colspan=""4"" width=""80"">&nbsp;</td></tr>

<tr>
<td colspan=""2"" width=""190"" align=left>**Sheet**Sheet</td>
<td colspan=""2"" width=""200"" align=left>**Rail**Rail</td>
<td colspan=""2"" width=""190"" align=left>**Reclamation**Reclamation
</td>
<td colspan=""4"" width=""80"">&nbsp;</td></tr>

<tr>
<td colspan=""2"" width=""190"" align=left>**TinPlate**Tin Plate</td>
<td colspan=""2"" width=""200"" align=left>**Bar**Bar</td>
<td colspan=""2"" width=""190"" align=left>**Secondary**Secondary
</td>
<td colspan=""4"" width=""80"">&nbsp;</td></tr>

<tr>
<td colspan=""10"" width=""650"">**Other**Other &nbsp; Specify:&nbsp;
&nbsp;**OtherSpec**</td>

<tr>
<td colspan=""10"" width=""650"">&nbsp;</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>Do you know which
Bethlehem business division will ship your order?</td></tr>

<tr>
<td colspan=""1"" width=""25"">&nbsp;</td>
<td colspan=""9"" width=""625"" align=left>**BusinessDivision**</td>
</tr>

<tr>
<td colspan=""10"" width=""650"">&nbsp;</td></tr>

<tr>
<td colspan=""10"" width=""650"">Is your order domestic or export?
&nbsp;&nbsp;&nbsp;**Domestic**</td></tr>
</table>

<table BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10>&nbsp;</td></tr>
<tr>
<td colspan=10 align=right>**GoToApp2**</td></tr>
<tr>
<td colspan=10 align=left>**BackToCreditHomePage**</td></tr>
</table>

```

SPIDERSESSION
[[/SPIDERPAGE]]

```

</BODY>
</HTML>

```

vii.

```

CreditDeptHomePg.htm
<HTML>
<HEAD>
<TITLE>CreditDept Home Page</TITLE>
</HEAD>

```

```

[[SPIDERPAGE NAME=CreditDeptHomePg]]

<center>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr><td colspan=10 align=center>**logobeth**</td></tr>
<tr><th colspan=10 align=center><font size=6>Credit Information
</font></th></tr>
<tr><td colspan=10 align=center bgcolor="#666699">&ampnbsp</td></tr>
<tr><td colspan=10 align=center>&ampnbsp</td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=500>
<tr><td align=center><font size=5><ul><li>**CreditInquiryPg**</li>
</ul></font></td></tr>
<tr><td align=left>-<i> A service for our customers whereby they can
provide access
by others to their credit history with Bethlehem.</i></td></tr>
</TABLE>

<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr><td colspan=10 align=center><hr width="650" color="black">
</td></tr>
<tr><th colspan=10 align=left><font size=4>New Customers</font></th>
</tr>
<tr>
<td colspan=10 align=left>
<ul>
<li>**CreditAppPgNewCust**</li>
<br>
<li>**CreditRepPgNewCust**</li>
</ul></td></tr>
<tr>
<td colspan=10 align=center><hr width="650" color="black"></td>
</tr>
<tr>
<td colspan=10 align=left>
<tr>
<th colspan=10 align=left><font size=4>Existing Customers </th></tr>
<tr>
<td colspan=10 align=left>
<ul>
<li>**WebOrderStatusLink** &ampnbsp&ampnbsp&ampnbsp (I.D. & Password for
'<i>Customer
Corner' required)</i></li>
<br>
<li>**CreditInstructions**</li>
</ul></td></tr>
<tr>
<td colspan=10 align=center><hr width="650" color="black"></td>
</tr>
<tr>
<td colspan=10 align=left>
<tr>
<th colspan=10 align=left><font size=4>Bethlehem Steel Policies</th>
</tr>
<tr>
<td colspan=10>
<ul>
<li>**BethCreditPolicy**</li>
<br>
<li>**BethDiscountPolicy**</li></ul></td></tr>
</TABLE>

<BR>**HomePage**

```

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

CreditDiscountPolicy.htm

<HTML>
<HEAD>
<TITLE>Credit Discount Policy</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditDiscountPolicy]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
<tr>
<td colspan=10 align=center> </td></tr>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color="white">Bethlehem's Standard Terms & Cash Discount Policy
</th></tr>
<tr>
<td colspan=10 align=left> </td></tr>
<tr>
<th colspan=10 align=left>Bethlehem Steel Corporation's standard,
unsecured / 'open', terms for
most products are
30-3/4-10. There are other unsecured terms that
are determined by product and
method of shipment.</th></tr>
<tr>
<td colspan=10 align=center> </td></tr>
<tr>
<th colspan=10 align=left>Payment, regardless of the specific terms,
is based on <u>receipt</u>
of your remittance. For example, on standard terms of 30-3/4-10, we
will consider a payment
"prompt" if the monies are received on or within 30 days from the
invoice date. Likewise, in
order to take advantage of the cash discount offered, payment must be
received within the period
specified in the terms. In the example above, receipt is required
within the 10 days of the
invoice date.</th></tr>
<tr>
<td colspan=10 align=center> </td></tr>
<tr>
<th colspan=10 align=left>For the convenience of our customers,
Bethlehem offers the option to
group invoices for payment (on the 'received' basis) twice per month.
</th></tr>
<tr>
<th colspan=10 align=left>Under these guidelines:</th></tr>
<tr>
<td colspan=10 align=left> </td></tr>

<tr>
<th colspan=10 align=left>
<dl>
<dt>On a Discount Basis -</dt>
<dd>Invoices dated from the 1st to the 15th - Payment due by the
25th of the same month.</dd>
<dd>Invoices dated from the 16th to the 31st - Payment due by the
10th of the following month.</dd></dl></th></tr>

```

<tr>
<th colspan=10 align=left>
<dl>
<dt>On a Net or ""Prompt"" Basis -</dt>
<dd>Invoices dated from the 1st to the 15th - Payment due by the
10th of the following month.</dd>
<dd>Invoices dated from the 16th to the 31st - Payment due by the
25th of the following month.</dd></dl></th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center><font size=3 color="#666699"><b>
QUESTIONS? email:</b>
<a href=""mailto:bscredit@bethsteel.com"">bscredit@bethsteel.com</a>
</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>**BackToCreditHomePage**</td></tr>

</TABLE>

```

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

ix.

CreditInquiry.htm

```

<HTML>
<HEAD>
<TITLE>Credit Inquiry</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditInquiry]]
<CENTER>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr></table>
<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<TH ALIGN=center VALIGN=MIDDLE bgcolor="#666699"><FONT SIZE=4
color="white"><B>Credit Inquiry</B></FONT></TH>
</TABLE>

<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<TH ALIGN=left VALIGN=left><FONT SIZE=3>As a matter of policy,
Bethlehem shares trade experience as a service to its customers.
</FONT></TH>
</TABLE>

<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<TH ALIGN=left VALIGN=left><FONT SIZE=3>This information is available
in an easy to use electronic format and can only
be accessed by obtaining a 6-digit code and 4-digit ""CIN"" number
from individual customers of Bethlehem.</FONT></TH>
</TABLE>

<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>

```

<TH ALIGN=left VALIGN=left>YOU, as a third party inquiring of Bethlehem's experience with one of our customers, will be allowed access to certain credit information. In order to access this information, you will need to enter the following information below:</TH>

</TABLE>

```
<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=50%>
<tr>
<TH ALIGN=left VALIGN=left>
<ol>
<li>Your company name/ address/ phone #<br><br>
<li>Your company's DUNS # and Fed I.D. #<br><br>
<li>Your name/ title<br><br>
<li>A six (6) digit code # supplied to you by our customer<br><br>
<li>A four (4) digit ""CIN"" (commercial identification number)
supplied to you by our customer.
</TH></tr>
<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<td colspan=""10"" align=center>**Continue**</td></tr>
<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<td colspan=""10"" align=center>**BackToCreditHomePage**</td></tr>
</TABLE>
```

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

x.

CreditInquiryForm.htm

<HTML>

<HEAD>

<TITLE>CreditInquiry Form</TITLE>

</HEAD>

[[SPIDERPAGE NAME=CreditInquiryForm]]

<center>

<TABLE BORDER=0 CELLSPACING=1 CELLPADDING=1 WIDTH=650>

<tr>

<td colspan=""10"" align=""center"">**logobeth**</td></tr>

<tr>

<td colspan=""10"" align=""center"">

ErrorMessage</td></tr>

</table>

<TABLE BORDER=0 CELLSPACING=1 CELLPADDING=1 WIDTH=650>

<TH ALIGN=LEFT COLSPAN=""10"" bgcolor=""#666699""> Enter your company name, address and phone number:</TH>

<TR>

<TD COLSPAN=""2""> </TD>

<TD COLSPAN=""3"">Company Name</TD>

<TD COLSPAN=""1""> </TD>

<TD colspan=""4"">**CpnyName**</TD></TR>

<TR>

```

<TD colspan=""2"">&nbsp;</td>
<TD align=left colspan=""3"">Street</td>
<TD colspan=""1"">&nbsp;</td>
<TD colspan=""4"">**CpnyStreet**</TD></TR>

<TR>
<TD colspan=""2"">&nbsp;</td>
<TD colspan=""3"" align=left>City </td>
<TD colspan=""1"">&nbsp;</td>
<TD colspan=""4"">**CpnyCity**</td></tr>

<tr>
<td colspan=""2"">&nbsp;</td>
<td colspan=""3"" align=left>State</td>
<td colspan=""1"">&nbsp;</td>
<td colspan=""4"">**CpnyState** &nbsp;&nbsp;Other&nbsp;**CpnyOther**</td></tr>

<tr>
<td colspan=""2"">&nbsp;</td>
<td colspan=""3"" align=left>Country</td>
<td colspan=""1"">&nbsp;</td>
<td colspan=""4"">**CpnyCountry** &nbsp;&nbsp;Zip Code&nbsp;&nbsp;**CpnyZip**</td></tr>

<TR>
<TD colspan=""2"">&nbsp;</td>
<TD align=left colspan=""3"">Phone Number</td>
<TD colspan=""1"">&nbsp;</td>
<td colspan=""4"">**CpnyPhone**</td></tr>

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<TH ALIGN=LEFT COLSPAN=""10"" bgcolor="#666699"><FONT SIZE=3 color="white">Enter your company's DUN'S # and Federal I.D.#
</font></TH></tr>
<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">DUNS #</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**Dn1**-**Dn2**-**Dn3**</td></tr>

<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"" align=left>Fed I.D.#
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**FdID1**-**FdID2**</td></tr>

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<TH ALIGN=LEFT COLSPAN=""10"" bgcolor="#666699"><FONT SIZE=3 color="white">Enter your name and title:</font></TH></tr>
<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">Name</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**CustName**</td></tr>

<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">Title</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**CustTitle**</td></tr>

```

```

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<th align=LEFT COLSPAN=""10"" bgcolor=""#666699""><font size=3
color=""white"">&nbsp;Enter the six (6) digit code # supplied by our
customer (subject company) to you:</font></th></tr>
<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">Code #</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**CstCde**</td></tr>

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<th align=LEFT COLSPAN=""10"" bgcolor=""#666699""><font size=3
color=""white"">&nbsp;Enter the four (4) digit ""CIN"" number
supplied by our customer (subject company) to you:</font></th></tr>
<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">CIN #</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**CINNum**</td></tr>
</table>

<br>
<table border=0 cellpadding=1 cellspacing=1 cellpadding=1 width=550 align=center>
<th align=left COLSPAN=""10""><font size=2>By clicking the button
below, you represent that the information retrieved will be
used only by your company, with no liability to Bethlehem Steel
Corporation, for
the sole purpose of assisting in <u>your</u> credit evaluation of
the subject company. In
addition, you represent that the subject company has specifically
given to you, or
your company, 'Bethlehem Steel Corporation' as a trade reference
for the purpose
of obtaining normal trade credit information.</font></th>
</tr></font></th>
<tr>
<td colspan=""10"" align=center>&nbsp;</td></tr>
<tr>
<td colspan=""3"">&nbsp;</td>
<td colspan=""2"" align=center>**Agree**</td>
<td colspan=""2"" align=center>**Decline**</td>
<td colspan=""3"">&nbsp;</td>
</tr>
<tr>
<td colspan=""10"" align=center>&nbsp;</td></tr>
<tr>
<td colspan=""10"" align=center>**BackToCreditHomePage**</td></tr>
</table>

<br>
<br>
<br>
<br>
**SPIDERSESSION**
[[/SPIDERPAGE]]

</body>
</html>

```

```
CreditInquiryResults.htm
<HTML>
<HEAD>
<TITLE>Credit Inquiry Results</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditInquiryResults]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
</table>

<br><br>

<TABLE BORDER=""0"" CELLSPACING=""0"" CELLPADDING=""0"" WIDTH=650>
<tr>
<td colspan=10 align=left bgcolor=""#666699""><font size=4
color=""white"">&nbsp;Your inquiry on : **CustName**&nbsp;
**CustLocation**</td></tr>
<tr>
<td colspan=10 align=left bgcolor=""#666699""><font size=4
color=""white"">
&nbsp;has produced the following history:</font></td></tr></table>
<TABLE BORDER=""0"" CELLSPACING=""0"" CELLPADDING=""0"" WIDTH=650>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=250>Years Sold&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;</th>
<td colspan=4 align=left width=250>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**YrsSold**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=250>High Credit&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;</th>
<td colspan=4 align=left width=250>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**HighCreidit**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=250>Terms&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;</th>
<td colspan=4 align=left width=250>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**Terms**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=300>Payment Days&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;</th>
<td colspan=4 align=left width=350>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**PaymentDays**</td></tr>
```

```

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=300>Last Sale&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;</th>
<td colspan=4 align=left width=350>&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**LastSale**</td></tr>

</TABLE>

<table border=0 cellpadding=2 cellspacing=2 width=650>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=5 align=left>**BackToCreditHomePage**</td>
<td colspan=5 align=right>**BackToInquiryForm**</td></tr>
</table>

```


****SPIDERSESSION****
[[/SPIDERPAGE]]

</BODY>
</HTML>

xii.

CreditPolicy.htm

```

<HTML>
<HEAD>
<TITLE>Credit Policy</TITLE>
</HEAD>
[[SPIDERPAGE NAME=CreditPolicy]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>

<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color="white">Bethlehem Steel Corporation Credit Policy</font></th>
</tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left>Bethlehem's Finance Department (Credit
Division) is responsible for
determining what constitutes an acceptable credit risk to Bethlehem
and for protecting its investment in Accounts Receivable consistent
with
Bethlehem's vision, strategies and objectives. It will develop
reasonable .

```

and prudent credit policies together with the Business Units and Field Sales to aid in maximizing consolidated Return on Net Assets. Credit exposure will be managed to minimize days sales outstanding (DSO) and bad debts.</th></tr>

```

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>**BackToCreditHomePage**</td></tr>

</TABLE>

```

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

xi.iii.

CreditRepNewCust.htm
<HTML>
<HEAD>
<TITLE>Credit Representative</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditRepNewCust]]

```

<center>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color="white">Contact Your Credit Representative</font></th></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left><u><b>We believe that communication is the
key to great customer
service.</u></b> Here is a tool to help you reach the credit person
within our
organization that handles your account.</td></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left>Bethlehem's Credit Division is a
decentralized function with credit
professionals located at each of our business divisions and at
corporate.</td></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left>Generally speaking, your company's
geographic location coupled with
the location of Bethlehem's business division that will produce and
ship your order, are the
criteria that determine the representative handling your account.
(If purchasing product
from more than one plant location, then the 'rule of thumb' defaults
to the representative
of the plant shipping the greater tonnage.)</td></tr>

```

```

<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left><hr width=""100%"" color=""black""></hr>
</td></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left><font size=4>New Customer</font></th></tr>
<tr>
<td colspan=10 align=left>If you are a prospective or new customer -
<b>CHOOSE</b> the geographic
area nearest you, product, and the Bethlehem <u>business division</u>
to produce your order:</td></tr>
<tr>
<td colspan=10 width=""650"">&nbsp;</td></tr>
<tr>
<td colspan=2 align=left><b>District</b>&nbsp;**District**&nbsp;
&nbsp;&nbsp;</td>
<td colspan=2 align=left><b>Product</b>&nbsp;</td>
<td colspan=6 align=left>**Product**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=left><b>Business Division</b>
&nbsp;&nbsp;**BusDivision**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=center>**CreditRepNewCustBttn**</td></tr>

<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>**GoToCreditHmPg**</td></tr>
</TABLE>

```

SPIDERSESSION
 [[/SPIDERPAGE]]

</BODY>
 </HTML>

xiv.

CreditRepNewResults.htm
 <HTML>
 <HEAD>
 <TITLE>Credit Representative Results</TITLE>
 </HEAD>
 [[SPIDERPAGE NAME=CreditRepNewResults]]
 <center>
 <TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
 <tr>
 <td colspan=10 align=center>**logobeth**</td></tr>
 </TABLE>

```

<TABLE BORDER=0 CELLPACING=2 CELLSPACING=2 WIDTH=650>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color=""white"">Your Credit Representative based on the information
entered is:</th></tr>
<tr><td colspan=10 align=left>&nbsp;</td></tr>
<tr><th colspan=10 align=center>**RepName**</th></tr>
<tr><th colspan=10 align=center>**RepTitle**</th></tr>
<tr><td colspan=10 align=center>&nbsp;</td></tr>
<tr><th colspan=10 align=center>Phone- **RepPhone**</th></tr>
<tr><th colspan=10 align=center>Fax- **RepFax**</th></tr>
<tr><td colspan=10 align=center>&nbsp;</td></tr>
<tr><td colspan=10 align=center><font size=2>To communicate via
email, click here:</font></td></tr>
<tr><th colspan=10 align=center><a href=""mailto:$$$$$$"><font
color=""blue"">**EmailCreditRep**</font></a></th></tr>
<tr><td colspan=10 align=center>&nbsp;</td></tr>
<tr><td colspan=5 align=left>**GoToCreditHmPg**</td>

<td colspan=5 align=right>**GoToCreditRepNewCust**</td></tr>
</table>

```

```

<BR>
**SPIDERSESSION**
[[/SPIDERPAGE]]

```

```

</BODY>
</HTML>

```

xv.

```

InquiryTrackScreen.htm
<HTML>
<HEAD>
<TITLE>Credit Inquiry Tracking Screen</TITLE>
</HEAD>

[[SPIDERPAGE NAME=InquiryTrackSceen]]
<center>
<TABLE BORDER=0 CELLPACING=2 CELLSPACING=2 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
<BR>
<tr>
<td align=center width=100%><font size=4>Credit Inquiry Tracking
Admin Screen</font></td></tr>
</table>

<TABLE BORDER=1 CELLSPACING=0 CELLSPACING=2 WIDTH=100%>
<tr>
<td align=center width=7% bgcolor="#666699"><font size=2
color=""white"">Customer Code</font></td>
<td align=center width=15% bgcolor="#666699"><font size=2
color=""white"">Company Name</font></td>
<td align=center width=20% bgcolor="#666699"><font size=2
color=""white"">User Name & Title</font></td>
<td align=center width=10% bgcolor="#666699"><font size=2
color=""white"">Phone Number</font></td>
<td align=center width=20% bgcolor="#666699"><font size=2

```

```

color=""white"">Address</font></td>
<td align=center width=4% bgcolor=""#666699""><font size=2
color=""white"">CIN Number</font></td>
<td align=center width=8% bgcolor=""#666699""><font size=2
color=""white"">Duns Number</font></td>
<td align=center width=8% bgcolor=""#666699""><font size=2
color=""white"">Fed ID</font></td>
<td align=center width=8% bgcolor=""#666699""><font size=2
color=""white"">Time Stamp</font></td></tr>

<BR><!-- [[SPIDERREPEAT NAME=Repeated1]] -->
<tr>
<td align=center width=7%><font size=2>**CusCode**</font></td>
<td align=center width=15%><font size=2>**CpnyNam**</font></td>
<td align=center width=20%><font size=2>**Nam**&nbsp;&nbsp;
**Tl**</font></td>
<td align=center width=10%><font size=2>**Ph**</font></td>
<td align=center width=20%><font size=2>**Str**&nbsp;**City**&nbsp;
**State**&nbsp;**Zip**&nbsp;**Country**&nbsp;**Other**</font></td>
<td align=center width=4%><font size=2>**CinNumber**</font></td>
<td align=center width=8%><font size=2>**Dn**</font></td>
<td align=center width=8%><font size=2>**Fd**</font></td>
<td align=center width=8%><font size=2>**TimeSt**</font></td>
</tr>

<!-- [[/SPIDERREPEAT]] -->
</table>

```

```

**SPIDERSESSION**
[[/SPIDERPAGE]]
</BODY>
</HTML>

```

xvi.

```

Override.htm
<HTML>
<HEAD>
<TITLE>OverrideTest</TITLE>
</HEAD>

[[SPIDERPAGE NAME=Override]]

<center>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr></table>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<th colspan=10 align=center bgcolor=""#666699""><font size=4
color=""white"">Override Payment Days</th></tr>
<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left width=650>&nbsp;&nbsp;&nbsp;Please
enter a Customer Code and the Payment Days override data:</th></tr>
<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left width=650>&nbsp;&nbsp;&nbsp;&nbsp;Customer
Code &nbsp;&nbsp;**CustCode2**&nbsp;&nbsp;Override Payment Days&nbsp;
&nbsp;**PaymentDays**
&nbsp;&nbsp;**Add***&nbsp;&nbsp;**Update***&nbsp;&nbsp;**Delete***</td>
</tr>

```

```

<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center width=650><font color=""red"">
**ErrorMessage**</font></td></tr>
<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<td colspan=5 align=left>**CreditHomePg**</td>
<td colspan=5 align=right>**WOSMainSel**</td></tr>

</TABLE>

```

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

xvii.

pgCreditInstruct.htm
<HTML>
<HEAD>
<TITLE>Credit Instructions</TITLE>
</HEAD>
[[SPIDERPAGE NAME=pgCreditInstruct]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
<tr><td colspan=10 align=center> </td></tr>
</TABLE>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=800>
<tr>
<td colspan=10 align=center bgcolor="#666699"> </td></tr>
<tr>
<td colspan=10 align=left> </td></tr>
<tr>
<th colspan=10 ><i>As an existing customer, if you
desire to use Bethlehem Steel as
a credit reference, you will
need to participate in our electronic credit system:</i></th>
</tr>
<tr>
<td colspan=10 align=left> </td></tr>
<tr><td colspan=10 align=center bgcolor="#666699"> </td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr><td colspan=10 align=left> </td></tr>
<tr><td colspan=10 align=left> </td></tr>
<tr>
<th colspan=10 align=left>(1) - Access 'Customer
Corner' (from Bethlehem's Home Page at

 www.bethsteel.com - 'Customers' - or use the link below.)
</th></tr>
<tr>

```

<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left><font size=4>(2) - Request Password (if
needed) by clicking on 'Need Password?'.
<br> After receiving your logon id and password via email,
return to the<br> Customer Corner as stated in (1) above.</font>
</th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<th colspan=10 align=left><font size=4>(3) - Click 'Customer Log In'.
</font>
</th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<th colspan=10 align=left><font size=4>(4) - Once you proceed beyond
the ""login"" screen, choose the link titled<br>
""Use Bethlehem as a Trade Reference"".</font>
</th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<th colspan=10 align=left><font size=4>(5) - Follow the instructions
for choosing and setting a ""CIN"" number.</font>
</th></tr>

</TABLE>

<br>
<br>
<table BORDER=0 CELLPACING=0 CELLSPACING=0 WIDTH=650>
<tr>
<td colspan=2 align=left><font color=blue>**GoToCustCorner**</font>
</td>
<td colspan=6 >&nbsp;&nbsp;</td>
<td colspan=2 align=right><font color=blue>**BackToCreditHomePage**</font>
</td>
</tr>
</TABLE>

<BR>
**SPIDERSESSION**
[[/SPIDERPAGE]]

</BODY>
</HTML>

```

xviii.

```

pgError.htm
<HTML>
<HEAD>
<TITLE>Error Handling</TITLE>
</HEAD>
<BODY>
<CENTER>
[[SPIDERPAGE NAME=pgError]]

<BR>
<table BORDER=0>
<tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color="#FF0000">
**TechnicalError**</font></td>
</tr><tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color="#FF0000">
**NoDataWarning**</font></td>
</tr><tr>

```

```

<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
**SqlServerError**</font></td>
</tr><tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
**LogonAgain**</font></td>
</tr><tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
</font></td>
</tr><tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
</font></td>
</tr>
</table>
<BR>
**SPIDERSESSION**
[ [ /SPIDERPAGE ] ]
</CENTER>
</font>
</BODY>
</HTML>

```

XII. JAVA Source Files

i.

```

AppProcessPg.java
// This file has been generated by NetDynamics Studio on Mon Jun. 19,
2000 16:03

```

```

package CreditApp;

import java.awt.event.*;
import spider.event.*;
import spider.database.*;
import spider.visual.*;

//THIS PAGE SHOWS UP IF THE USER'S APPLICATION WAS SUCESSFULLY SENT
TO THE EMAIL ACCOUNT

//[[SPIDER_CLASS BEGIN
public class AppProcessPg extends spider.visual.CSpPage
//]]SPIDER_CLASS END
{
    //[[SPIDER_EVENTS BEGIN
        //[[SPIDER_EVENT<GoToCreditHmPg_onWebEvent>
        public int GoToCreditHmPg_onWebEvent (CSpWebEvent
event)
        {
            return load("'"CreditDeptHomePg'");
        }
    //]]SPIDER_EVENT<GoToCreditHmPg_onWebEvent>
    //]]SPIDER_EVENTS END
}

```

ii.

```

AppResults.java
// This file has been generated by NetDynamics Studio on Thu Jun. 15,
2000 08:50

```

```

package CreditApp;

import java.awt.event.*;
import spider.event.*;

```

```

import spider.database.*;
import spider.visual.*;
import spider.util.*;
import spider.*;
import spider.session.*;
import java.lang.*;

//[[SPIDER_CLASS BEGIN
public class AppResults extends spider.visual.CSpPage
//]]SPIDER_CLASS END
{
    private transient Credit the_CreditObject;
    //[[SPIDER_EVENTS BEGIN

        //[[SPIDER_EVENT<this_onBeforeLoadEvent>
        public int this_onBeforeLoadEvent(CSpPageEvent event)
        {
            the_CreditObject = (Credit)
Credit.getInstance();
            return (PROCEED);
        }
    //]]SPIDER_EVENT<this_onBeforeLoadEvent>

    //[[SPIDER_EVENT
<IncorporateName_onBeforeDisplayEvent>
        public int IncorporateName_onBeforeDisplayEvent
(CSpDisplayEvent event)
        {
            //The code below(all of it) checks to see if the
            field is null, if it is then the
                //field will be displayed on the AppResults page for
            the user to fill in again.

            int x =
the_CreditObject.getIncorporatedName().trim().length();
            if( x != 0)
                return SKIP;
            else
                return PROCEED;

        }
    //]]SPIDER_EVENT
<IncorporateName_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<CpnyNameLabel_onBeforeDisplayEvent>
        public int CpnyNameLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
        {
            int x =
the_CreditObject.getIncorporatedName().trim().length();
            if( x != 0)
                return SKIP;
            else
                return PROCEED;
        }
    //]]SPIDER_EVENT<CpnyNameLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<IncorpName_onBeforeDisplayEvent>
        public int IncorpName_onBeforeDisplayEvent
(CSpDisplayEvent event)
        {
            int x =

```

```

the_CreditObject.getIncorporatedName().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<IncorpName_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadAddLabel_onBeforeDisplayEvent>
public int HeadAddLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getHeadquarterAddress().trim().length();
    if( a != 0)
    {
        int b =
the_CreditObject.getHeadquarterCity().trim().length();
        if( b != 0)
        {
            int c =
the_CreditObject.getHeadquarterState().trim().length();
            if( c != 0)
            {
                int d =
the_CreditObject.getHeadquarterZip().trim().length();
                if( d != 0)
                {
                    int e =
the_CreditObject.getHeadquarterPhone().trim().length();
                    if( e != 0)
                    {
                        return SKIP;
                    }
                    else
                    {
                        return PROCEED;
                    }
                }
                else
                {
                    return PROCEED;
                }
            }
            else
            {
                return PROCEED;
            }
        }
        else
        {
            return PROCEED;
        }
    }
    else
    {
        return PROCEED;
    }
}
//]]SPIDER_EVENT<HeadAddLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadStAdd_onBeforeDisplayEvent>
public int HeadStAdd_onBeforeDisplayEvent
(CSpDisplayEvent event)

```

```

    {
        int x =
the_CreditObject.getHeadquarterAddress().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
    }
//]]SPIDER_EVENT<HeadStAdd_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadStLabel_onBeforeDisplayEvent>
public int HeadStLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    ,
        int x =
the_CreditObject.getHeadquarterAddress().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
}
//]]SPIDER_EVENT<HeadStLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquCity_onBeforeDisplayEvent>
public int HeadquCity_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    ,
        int x =
the_CreditObject.getHeadquarterCity().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
}
//]]SPIDER_EVENT<HeadquCity_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadCityLabel_onBeforeDisplayEvent>
public int HeadCityLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    ,
        int x =
the_CreditObject.getHeadquarterCity().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
}
//]]SPIDER_EVENT<HeadCityLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadState_onBeforeDisplayEvent>
public int HeadState_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    ,
        int x =
the_CreditObject.getHeadquarterState().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
}
//]]SPIDER_EVENT<HeadState_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquStLabel_onBeforeDisplayEvent>
public int HeadquStLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)

```

```

        {
            int x =
the_CreditObject.getHeadquarterState().trim().length();
            if( x != 0)
                return SKIP;
            else
                return PROCEED;
        }
//]]SPIDER_EVENT<HeadquStLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadZip_onBeforeDisplayEvent>
public int HeadZip_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterZip().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadZip_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquZipLabel_onBeforeDisplayEvent>
public int HeadquZipLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterZip().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadquZipLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquPh_onBeforeDisplayEvent>
public int HeadquPh_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterPhone().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadquPh_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadPhLabel_onBeforeDisplayEvent>
public int HeadPhLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterPhone().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadPhLabel_onBeforeDisplayEvent>
```

```

//[[SPIDER_EVENT<DunFedLabel_onBeforeDisplayEvent>
public int DunFedLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
().length() !=0 &&
().length() !=0)
    {
        if(the_CreditObject.getFedID1
().trim().length() !=0 &&
().trim().length() !=0 )
            return SKIP;
        else
            return (PROCEED);
    }
    else
        return (PROCEED);
}
//]]SPIDER_EVENT<DunFedLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Duns1_onBeforeDisplayEvent>
public int Duns1_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
().length() !=0 &&
().length() !=0)
        return SKIP;
    else
        return (PROCEED);
}
//]]SPIDER_EVENT<Duns1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Duns2_onBeforeDisplayEvent>
public int Duns2_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
().length() !=0 &&
().length() !=0)
        return SKIP;
    else
        return (PROCEED);
}
//]]SPIDER_EVENT<Duns2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Duns3_onBeforeDisplayEvent>
public int Duns3_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
the_CreditObject.getDun2().trim

```

```

().length() !=0 &&
                           the_CreditObject.getDun3().trim
().length() !=0)
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<Duns3_onBeforeDisplayEvent>

//[[SPIDER_EVENT<DunsLabel_onBeforeDisplayEvent>
public int DunsLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
if(the_CreditObject.getDun1().trim
().length() !=0 &&
                           the_CreditObject.getDun2().trim
().length() !=0 &&
                           the_CreditObject.getDun3().trim
().length() !=0)
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<DunsLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Fed1_onBeforeDisplayEvent>
public int Fed1_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
if(the_CreditObject.getFedID1().trim
().length() !=0 &&
                           the_CreditObject.getFedID2().trim
().length() !=0 )
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<Fed1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Fed2_onBeforeDisplayEvent>
public int Fed2_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
if(the_CreditObject.getFedID1().trim
().length() !=0 &&
                           the_CreditObject.getFedID2().trim
().length() !=0 )
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<Fed2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<FedLabel_onBeforeDisplayEvent>
public int FedLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
if(the_CreditObject.getFedID1().trim
().length() !=0 &&
                           the_CreditObject.getFedID2().trim
().length() !=0 )
                           return SKIP;
else
                           return (PROCEED);
}

```

```

        }
    //]]SPIDER_EVENT<FedLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankRefLabel_onBeforeDisplayEvent>
    public int BankRefLabel_onBeforeDisplayEvent
    (CSpDisplayEvent event)
    {
        int a = the_CreditObject.getBankName
        ().trim().length();
        if( a != 0)
        {
            int b =
            the_CreditObject.getBankStreet().trim().length();
            if( b != 0)
            {
                int c =
                the_CreditObject.getBankCity().trim().length();
                if( c != 0)
                {
                    int d =
                    the_CreditObject.getBankState().trim().length();
                    if( d != 0)
                    {
                        int e =
                        the_CreditObject.getBankZip().trim().length();
                        if( e != 0)
                            return SKIP;
                        else
                            return PROCEED;
                    }
                    else
                        return PROCEED;
                }
                else
                    return PROCEED;
            }
            else
                return PROCEED;
        }
        else
            return PROCEED;
    }
    //]]SPIDER_EVENT<BankRefLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankName1_onBeforeDisplayEvent>
    public int BankName1_onBeforeDisplayEvent
    (CSpDisplayEvent event)
    {
        int x = the_CreditObject.getBankName
        ().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
    }
    //]]SPIDER_EVENT<BankName1_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankNameLb_onBeforeDisplayEvent>
    public int BankNameLb_onBeforeDisplayEvent
    (CSpDisplayEvent event)
    {
        int x = the_CreditObject.getBankName
        ().trim().length();
        if( x != 0)

```

```

                return SKIP;
            else
                return PROCEED;
        }
    //]]SPIDER_EVENT<BankNameLb_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankStLabel_onBeforeDisplayEvent>
    public int BankStLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getBankStreet().trim().length();
    if( x != 0 )
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankStLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankStr_onBeforeDisplayEvent>
    public int BankStr_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getBankStreet().trim().length();
    if( x != 0 )
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankStr_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankCity1_onBeforeDisplayEvent>
    public int BankCity1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankCity
().trim().length();
    if( x != 0 )
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankCity1_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankCityLb_onBeforeDisplayEvent>
    public int BankCityLb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankCity
().trim().length();
    if( x != 0 )
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankCityLb_onBeforeDisplayEvent>

```

```

        //[[SPIDER_EVENT<BankSt1_onBeforeDisplayEvent>
        public int BankSt1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankState
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankSt1_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<BankStLb_onBeforeDisplayEvent>
        public int BankStLb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankState
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankStLb_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<BankZip1_onBeforeDisplayEvent>
        public int BankZip1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankZip
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankZip1_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<BankZipLb_onBeforeDisplayEvent>
        public int BankZipLb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankZip
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankZipLb_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<TradeRefLabel_onBeforeDisplayEvent>
        public int TradeRefLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName1().trim().length();
    if( a != 0)
    {
        int b =
the_CreditObject.getTradeRefCityStZip1().trim().length();

```

```

        if( b != 0 )
        {
            int c =
the_CreditObject.getTradeRefName2().trim().length();
            if( c != 0 )
            {
                int d =
the_CreditObject.getTradeRefCityStZip2().trim().length();
                if( d != 0 )
                {
                    int e =
the_CreditObject.getTradeRefName3().trim().length();
                    if( e != 0 )
                    {
                        int f =
the_CreditObject.getTradeRefCityStZip3().trim().length();
                        if( f != 0 )
                            return SKIP;
                        else
                            return
PROCEED;
                    }
                    else
                        return PROCEED;
                }
                else
                    return PROCEED;
            }
            else
                return PROCEED;
        }
        else
            return PROCEED;
    }
//]]SPIDER_EVENT<TradeRefLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref1Lab_onBeforeDisplayEvent>
public int Ref1Lab_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName1().trim().length();
    if( a != 0 )
    {
        int b =
the_CreditObject.getTradeRefCityStZip1().trim().length();
        if( b != 0 )
            return SKIP;
        else
            return PROCEED;
    }
    else
        return PROCEED;
}
//]]SPIDER_EVENT<Ref1Lab_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref1NameLabel_onBeforeDisplayEvent>
public int Ref1NameLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =

```

```

the_CreditObject.getTradeRefName1().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefName1_onBeforeDisplayEvent>
public int RefName1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getTradeRefName1().trim
().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref1CityLabel_onBeforeDisplayEvent>
public int Ref1CityLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip1().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefCity1_onBeforeDisplayEvent>
public int RefCity1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip1().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref2Lab_onBeforeDisplayEvent>
public int Ref2Lab_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName2().trim().length();
    if( a != 0)
    {
        int b =
the_CreditObject.getTradeRefCityStZip2().trim().length();
        if( b != 0)
            return SKIP;
        else
            return PROCEED;
    }
    else

```

```

                                return PROCEED;;
}
//]]SPIDER_EVENT<Ref2Lab_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref2NameLabel_onBeforeDisplayEvent>
public int Ref2NameLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefName2().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefName2_onBeforeDisplayEvent>
public int RefName2_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefName2().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref2CityLabel_onBeforeDisplayEvent>
public int Ref2CityLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip2().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefCity2_onBeforeDisplayEvent>
public int RefCity2_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip2().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref3Lab_onBeforeDisplayEvent>
public int Ref3Lab_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName3().trim().length();
    if( a != 0)

```

```

        {
            int b =
the_CreditObject.getTradeRefCityStZip3().trim().length();
            if( b != 0)
                return SKIP;
            else
                return PROCEED;
        }
    else
        return PROCEED;
}
//]]SPIDER_EVENT<Ref3Lab_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref3NameLabel_onBeforeDisplayEvent>
public int Ref3NameLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefName3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName3_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefName3_onBeforeDisplayEvent>
public int RefName3_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefName3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName3_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref3CityLabel_onBeforeDisplayEvent>
public int Ref3CityLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity3_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefCity3_onBeforeDisplayEvent>
public int RefCity3_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity3_onBeforeDisplayEvent>
```

```

//[[SPIDER_EVENT<Submit_onWebEvent>
public int Submit_onWebEvent(CSpWebEvent event)
{
    Credit the_CreditObject =
Credit.getInstance();

        /* setting all the fields on this page so when
the user hits the ""submit"" button
                their information is saved */
        the_CreditObject.setIncorporateName
(getDisplayField("IncorporateName").getValue().toString());
        the_CreditObject.setHeadStAdd
(getDisplayField("HeadStAdd").getValue().toString());
        the_CreditObject.setHeadquCity
(getDisplayField("HeadquCity").getValue().toString());
        the_CreditObject.setHeadState(getDisplayField
("HeadState").getValue().toString());
        the_CreditObject.setHeadZip
(getDisplayField("HeadZip").getValue().toString());
        the_CreditObject.setHeadquPh
(getDisplayField("HeadquPh").getValue().toString());
        //            the_CreditObject.setHeadquFax
(getDisplayField("HeadquFax").getValue().toString());
        //            the_CreditObject.setHeadquWww
(getDisplayField("HeadquWww").getValue().toString());
        the_CreditObject.setDuns1
(getDisplayField("Duns1").getValue().toString());
        the_CreditObject.setDuns2
(getDisplayField("Duns2").getValue().toString());
        the_CreditObject.setDuns3
(getDisplayField("Duns3").getValue().toString());
        the_CreditObject.setFed1
(getDisplayField("Fed1").getValue().toString());
        the_CreditObject.setFed2
(getDisplayField("Fed2").getValue().toString());
        the_CreditObject.setBankName1
(getDisplayField("BankName1").getValue().toString());
        the_CreditObject.setBankStr
(getDisplayField("BankStr").getValue().toString());
        the_CreditObject.setBankCity1
(getDisplayField("BankCity1").getValue().toString());
        the_CreditObject.setBankSt1
(getDisplayField("BankSt1").getValue().toString());
        the_CreditObject.setBankzip1
(getDisplayField("BankZip1").getValue().toString());
        //            the_CreditObject.setAcctNm
(getDisplayField("AcctNm").getValue().toString());
        //            the_CreditObject.setAcctPhone
(getDisplayField("AcctPhone").getValue().toString());
        the_CreditObject.setRefName1
(getDisplayField("RefName1").getValue().toString());
        //            the_CreditObject.setRefStreet1
(getDisplayField("RefStreet1").getValue().toString());
        the_CreditObject.setRefCity1
(getDisplayField("RefCity1").getValue().toString());
        //            the_CreditObject.setAttnName1
(getDisplayField("AttnName1").getValue().toString());
        //            the_CreditObject.setRefPh1
(getDisplayField("RefPh1").getValue().toString());
        //            the_CreditObject.setRefFaxNum1
(getDisplayField("RefFaxNum1").getValue().toString());

```

```

        the_CreditObject.setRefName2
(getDisplayField("RefName2").getValue().toString());
//          the_CreditObject.setRefStreet2
(getDisplayField("RefStreet2").getValue().toString());
//          the_CreditObject.setRefCity2
(getDisplayField("RefCity2").getValue().toString());
//          the_CreditObject.setAttnName2
(getDisplayField("AttnName2").getValue().toString());
//          the_CreditObject.setRefPh2
(getDisplayField("RefPh2").getValue().toString());
//          the_CreditObject.setRefFaxNum2
(getDisplayField("RefFaxNum2").getValue().toString());
//          the_CreditObject.setRefName3
(getDisplayField("RefName3").getValue().toString());
//          the_CreditObject.setRefStreet3
(getDisplayField("RefStreet3").getValue().toString());
//          the_CreditObject.setRefCity3
(getDisplayField("RefCity3").getValue().toString());
//          the_CreditObject.setAttnName3
(getDisplayField("AttnName3").getValue().toString());
//          the_CreditObject.setRefPh3
(getDisplayField("RefPh3").getValue().toString());
//          the_CreditObject.setRefFaxNum3
(getDisplayField("RefFaxNum3").getValue().toString());

//                                if(! the_CreditObject.isValid())
//                                return load("AppResults");

//BethSteel set up this email account
String to =
""bscredit@bethsteel.com";
//      String to = ""vicky.alban@eds.com";
String from = getFieldValue
("From").toString();
if(from.indexOf("@",0) == -1 ||
from.indexOf(".",0) == -1)
{
    from =
"mailServer@bsco.com";
}
String host = "mail.bsco.com";
String host = "10.50.1.1";
String subject = "Process Credit
Application";

//this is the code that will be
displayed on the emial sent to the above address
String msgText = "" Past Customer:
"" + the_CreditObject.getPastCust()           +
""\n What Year: ""
+ the_CreditObject.getPastYear()               +
""\n Name Used: ""
+ the_CreditObject.getPastName()              +
""\n Product: "" +
the_CreditObject.getPlate() + "" +
the_CreditObject.getSemiFinished() + "" +
the_CreditObject.getPipe() + "" +
the_CreditObject.getSheet() + "
" +
the_CreditObject.getReclamation() + "" +
the_CreditObject.getTinPlate() + "" +
the_CreditObject.getBar() + "
" +
the_CreditObject.getSecondary() + "" +
the_CreditObject.getOther() +
"" +
the_CreditObject.getOtherSpec()

```

```

+ """\n Business   +
Division: "" + the_CreditObject.getBusinessDivision()
+ """\n Domestic: "" +
the_CreditObject.getDomestic()
+ """\n"""
+ """\n Company
Information: """
+ """\n Company Name:
"" + the_CreditObject.getIncorporatedName() +
the_CreditObject.getIncorporateName()
+ """\n Company
Division Name: "" + the_CreditObject.getCpnyDivisionName() + """\n PO
Name: "" + the_CreditObject.getPOName()
+ """\n"""
+ """\n Headquarter
Address: """
+ """\n Company
Headquarter Street Address: "" +
the_CreditObject.getHeadquarterAddress() +
the_CreditObject.getHeadStAdd()
+ """\n Company
Headquarter Box Number: "" + the_CreditObject.getHeadquarterBoxNum()
+ """\n Company
Headquarter City: "" + the_CreditObject.getHeadquarterCity() +
the_CreditObject.getHeadquCity()
+ """\n Company
Headquarter Zip: "" + the_CreditObject.getHeadquarterZip() +
the_CreditObject.getHeadZip()
+ """\n Company
Headquarter State: "" + the_CreditObject.getHeadquarterState() +
the_CreditObject.getHeadState()
+ """\n Company
Headquarter Foreign Address (if applicable): "" +
the_CreditObject.getHeadquarterForeignAddress()
+ """\n Company
Headquarter Phone Number: "" + the_CreditObject.getHeadquarterPhone()
+ the_CreditObject.getHeadquPh()
+ """\n Company
Headquarter Fax Number: "" + the_CreditObject.getHeadquarterFax() +
the_CreditObject.getHeadquFax()
+ """\n Company
Headquarter Web Address: "" + the_CreditObject.getHeadquarterWeb() +
the_CreditObject.getHeadquWww()
+ """\n"""
+ """\n Branch
Location Information: """
+ """\n Will they
Purchase through a branch location?: "" +
the_CreditObject.getPurchaseThroughBranch()
+ """\n Branch Name:
"" + the_CreditObject.getBranchName() + """\n Branch Division: "" +
the_CreditObject.getBranchDivision()
+ """\n Branch Street
Address: "" + the_CreditObject.getBranchStAddress()
+ """\n Branch City:
"" + the_CreditObject.getBranchCity() + """\n Branch State: "" +
the_CreditObject.getBranchState()
+ """\n Branch Zip:
"" + the_CreditObject.getBranchZip() + """\n Branch Foreign Address:
"" + the_CreditObject.getBranchForeignAddress()
+ """\n Branch Phone
Number: "" + the_CreditObject.getBranchPhone() + """\n Branch Fax
Number: "" + the_CreditObject.getBranchFax()
//+ """\n Branch
Street Address: "" + the_CreditObject.getBranchStAddress()
+ """\n"""

```

```

+ """\n Invoices and
Other Correspondence Information: """
+ """\n Invoices
should be sent to: "" + the_CreditObject.getInvoiceCorrespondence()
+ """\n Other
locations Street Address: "" + the_CreditObject.getOtherStreet()
+ """\n Other
locations Box Number: "" + the_CreditObject.getOtherBoxNum()
+ """\n Other
locations City: "" + the_CreditObject.getOtherCity() + """\n Other
locations state: "" + the_CreditObject.getOtherState()
+ """\n Other
locations Zip Code: "" + the_CreditObject.getOtherZip() + """\n
Foreign Address if applicable: "" +
the_CreditObject.getInvoiceForeignAddress()
+ """\n"""
+ """\n Bethlehem

Sales Person Information: """
+ """\n Have you ever
contacted Bethlehem in the past?: "" +
the_CreditObject.getContactWithBeth()
+ """\n If yes, please
provide the Contacts Name: "" + the_CreditObject.getContactBethName()
+ """\n"""
+ """\n Accounts

Payable Information: """
+ """\n Accounts
Payable system is: "" + the_CreditObject.getApSystem() + """\n
Payments are mailed to: "" + the_CreditObject.getApMailedTo()
+ """\n Buyer's Name:
"" + the_CreditObject.getBuyersName() + """\n Buyer's Phone Number:
"" + the_CreditObject.getBuyersPhoneNum()
+ """\n Buyer's Email
Address: "" + the_CreditObject.getBuyersEmail() + """\n AP Supervisors
Name: "" + the_CreditObject.getApSupervisorName()
+ """\n AP Supervisors
Phone Number: "" + the_CreditObject.getApPhoneNum() + """\n AP
Supervisors Email Address: "" + the_CreditObject.getApEmail()
+ """\n Person to
contact for a financial statement: "" +
the_CreditObject.getFinancialSupportName()
+ """\n"""
+ """\n DUNS # and Fed

I.D. Information: """
+ """\n DUNS Number:
"" + the_CreditObject.getDun1() + """-""" + the_CreditObject.getDun2()
+ """-""" + the_CreditObject.getDun3() + the_CreditObject.getDuns1() +
"""-""" + the_CreditObject.getDuns2() + """-""" +
the_CreditObject.getDuns3()
+ """\n Fed ID: "" +
the_CreditObject.getFedID1() + """-""" + the_CreditObject.getFedID2() +
the_CreditObject.getFed1() + """-""" + the_CreditObject.getFed2()
+ """\n"""
+ """\n Primary Bank

Reference Information: """
+ """\n Bank Name: """
+ the_CreditObject.getBankName() + the_CreditObject.getBankName1() +
"""\n Bank Street Address: "" + the_CreditObject.getBankStreet() +
the_CreditObject.getBankStr()
+ """\n Bank City: """
+ the_CreditObject.getBankCity() + the_CreditObject.getBankCity1() +
"""\n Bank State: "" + the_CreditObject.getBankState() +
the_CreditObject.getBankSt1()
+ """\n Banks Country:
"" + the_CreditObject.getBankOther() + """\n Bank Zip Code: "" +
the_CreditObject.getBankZip() + the_CreditObject.getBankZip1()

```

```

        + """\n Account
Officers Name: "" + the_CreditObject.getAcctOfficerName() +
the_CreditObject.getAcctNm() + ""\n Account Officers Phone Number:
"" + the_CreditObject.getAcctOfficerPhoneNum() +
the_CreditObject.getAcctPhone()
        + """\n"""
        + """\n Trade

Reference Information: """
        + """\n Trade
Reference #1 Name: "" + the_CreditObject.getTradeRefName1() +
the_CreditObject.getRefName1() + ""\n Trade Reference #1 Street
Address: "" + the_CreditObject.getTradeRefStreet1() +
the_CreditObject.getRefStreet1()
        + """\n Trade

Reference #1 City State and Zip: "" +
the_CreditObject.getTradeRefCityStZip1() +
the_CreditObject.getRefCity1() + ""\n Trade Reference #1 Attn: "" +
the_CreditObject.getTradeRefAttn1() + the_CreditObject.getAttnName1()
        + """\n Trade

Reference #1 Phone Number: "" + the_CreditObject.getTradeRefPhone1()
+ the_CreditObject.getRefPh1() + ""\n Trade Reference #1 Fax Number:
"" + the_CreditObject.getTradeRefFax1() +
the_CreditObject.getRefFaxNum1()
        + """\n Trade

Reference #2 Name: "" + the_CreditObject.getTradeRefName2() +
the_CreditObject.getRefName2() + ""\n Trade Reference #2 Street: "" +
the_CreditObject.getTradeRefStreet2() +
the_CreditObject.getRefStreet2()
        + """\n Trade

Reference #2 City State and Zip: "" +
the_CreditObject.getTradeRefCityStZip2() +
the_CreditObject.getRefCity2() + ""\n Trade Reference #2 Attn: "" +
the_CreditObject.getTradeRefAttn2() + the_CreditObject.getAttnName2()
        + """\n Trade

Reference #2 Phone Number: "" + the_CreditObject.getTradeRefPhone2()
+ the_CreditObject.getRefPh2() + ""\n Trade Reference #2 Fax Number:
"" + the_CreditObject.getTradeRefFax2() +
the_CreditObject.getRefFaxNum2()
        + """\n Trade

Reference #3 Name: "" + the_CreditObject.getTradeRefName3() +
the_CreditObject.getRefName3() + ""\n Trade Reference #3 Street: "" +
the_CreditObject.getTradeRefStreet3() +
the_CreditObject.getRefStreet3()
        + """\n Trade

Reference #3 City State and Zip: "" +
the_CreditObject.getTradeRefCityStZip3() +
the_CreditObject.getRefCity3() + ""\n Trade Reference #3 Attn: "" +
the_CreditObject.getTradeRefAttn3() + the_CreditObject.getAttnName3()
        + """\n Trade

Reference #3 Phone Number: "" + the_CreditObject.getTradeRefPhone3()
+ the_CreditObject.getRefPh3() + ""\n Trade Reference #3 Fax Number:
"" + the_CreditObject.getTradeRefFax3() +
the_CreditObject.getRefFaxNum3();

```

```

//myCart =
Cart.getCartFromUserSession();
//Hashtable userCart = myCart.getCart
();
//String[] userItem;

/*for (Enumeration e =
userCart.elements(); e.hasMoreElements();) */

```

```

        {
            userItem = (String[]) e.nextElement
        );
        for(int i=0;i<userItem.length;i++)
        {
            if
            (userItem[i].equals(""\&nbsp;""))
            userItem[i] = """";
        }
        msgText = msgText +
        """\n          Key: "" + userItem[0]
        + """\n
        Quarter: "" + userItem[1]
        + """\n
        Year: "" + userItem[2]
        + """
        ""\nAuthor/Inventor: "" + userItem[3] + """\n          Title: "" +
        userItem[4]
        + """\n          Patent
        No: "" + userItem[5] + """\n\n""";
    }*/
    EmailObj the_Mailer = new EmailObj();
    the_Mailer.sendMail(host, from, to,
    subject, msgText);

    Credit.saveInstance
    (the_CreditObject);
    //return(command);
    return load("AppProcessPg");
}
//]] SPIDER_EVENT<Submit_onWebEvent>

//]] SPIDER_EVENTS END
}

iii.

CBethIMSDataField.java
package ImsUtilities;

import spider.database.*;
import spider.session.*;
import java.lang.*;
import spider.util.*;

public abstract class CBethIMSDataField extends CSpDataField
{
    public static final boolean LOCAL_DEBUG = false;
    public static final int NO_PADDING = 0;
    public static final int FILLER_ONLY = 1;
    public static final int PAD_LEFT = 2;
    public static final int PAD_RIGHT = 3;
    public static final int NO_CASE_CHANGE = 0;
    public static final int UPPER_CASE = 1;
    public static final int LOWER_CASE = 2;
    public static final char DEFAULT_FILLER_CHARACTER =

```

```

    ' ';

        protected void invalidateField(String pReason)
        {
            CBethIMSProcedure the_parent =
(CBethIMSProcedure)getParent();
            CSpLog.send
(this,CSpLog.WARNING,pReason);
            the_parent.invalidateParameters();
        }

        public int afterInit()
        {
            int command = super.afterInit();
            if(PROCEED != command)
                return command;
            //
            // Override the Studio Properties
            //
            setOnEmptyValue
(ON_EMPTY_VALUE_EXCLUDE);
            return PROCEED;
        }

        public abstract int getPaddingType();
        public abstract int getCaseChange();

        public char getFillerCharacter()
        {
            return DEFAULT_FILLER_CHARACTER;
        }

        //
        // derived classes MAY override this method to
provide specialized behavior
        // which CSpDataField and CBethIMSDataField do not
provide
        //
        protected CSpValue getIMSValue()
        {
            if(LOCAL_DEBUG)
                CSpLog.send
(this,CSpLog.USER_DEBUG,""CBethIMSDataField.getIMSValue() entering
super.getValue()"");
            CSpValue the_value = super.getValue
();
            if(LOCAL_DEBUG)
                CSpLog.send
(this,CSpLog.USER_DEBUG,""CBethIMSDataField.getIMSValue() return from
super.getValue()"");
            return the_value;
        }

        //
        // This method is final so that derived classes
enforce the mandatory
        // formatting which CBethIMSDataField provides.
getValue() behavior MAY
        // be directed by overriding the getFillerCharacter()
method and MUST be
        // directed by providing concrete method
getPaddingType()
        //
        // The getValue() behavior may be skipped entirely by
having method
        // getPaddingType() return

```

```

CBethIMSDataField.NO_PADDING
    //
    public final CSpValue getValue()
    {
        if(LOCAL_DEBUG)           CSpLog.send
(this,CSpLog.USER_DEBUG,""BEGIN Data Object [" + getParent().getName()
+ ""] getValue()"" + getName());      if(!(((CBethIMSPprocedure)getParent
()).areParametersValid()))
        {
            //
            // if someone else
already invalidated then stop processing
            //
            return new CSpNull();
        }
        int the_padding_type = getPaddingType
();
        int the_case_change = getCaseChange
();
        //
        // Allow skipping default behavior
        //
        if( (NO_PADDING == the_padding_type)
&&
                (NO_CASE_CHANGE ==
the_case_change) )
        {
            if(LOCAL_DEBUG)
CSpLog.send(this,CSpLog.USER_DEBUG,""NO_PADDING and
NO_CASE_CHANGE"");
                if(LOCAL_DEBUG)
CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue()"");
                    return getIMSValue();
            }
            else if( (NO_PADDING ==
the_padding_type) &&
                the_case_change) )
            {
                if(LOCAL_DEBUG)
CSpLog.send(this,CSpLog.USER_DEBUG,""JUST LOWER CASE"");
                    CSpValue the_value =
getIMSValue();
                if(LOCAL_DEBUG)
CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue()"");
                    return new CSpString
(the_value.stringValue().toLowerCase());
            }
            else if( (NO_PADDING ==
the_padding_type) &&
                the_case_change) )
            {
                if(LOCAL_DEBUG)
CSpLog.send(this,CSpLog.USER_DEBUG,""JUST UPPER CASE"");
                    CSpValue the_value =
getIMSValue();
                if(LOCAL_DEBUG)

```

```

CSpLog.send(this,CSpLog.USER_DEBUG, ""END getValue() "");
                                                 return new CSpString
(the_value.stringValue().toUpperCase());
}

// Allocate buffer of exact size
int the_column_length =
getColumnLength();
StringBuffer the_string_buffer = new
StringBuffer(the_column_length);
char the_filler_character =
getFillerCharacter();

if(LOCAL_DEBUG)
CSpLog.send
(this,CSpLog.USER_DEBUG, "ColumnLength " + the_column_length);

switch (the_padding_type)
{
    case FILLER_ONLY :
    {
        if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG, ""FILLER_ONLY");
for
(int i=0;i<the_column_length;i++)
the_string_buffer.append(the_filler_character);

break;
    }
    case PAD_LEFT :
    case PAD_RIGHT :
    {

CSpValue the_value = getIMSValue();
if(!
(((CBethIMSProcedure)getParent()).areParametersValid()))
{

// if getIMSValue() invalidated the field
//

return new CSpNull();
}
if
(null == the_value)
{
invalidateField("CBethIMSDataField does not allow getIMSValue() =
null, returning CSpNull to get past");

return new CSpNull();
}
if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG, "getIMSValue() = " +
the_value.stringValue());

```



```

CSpLog.send(this,CSpLog.WARNING,
""Invalid Padding Type "" + the_padding_type);

return getIMSValue();
}

if(LOWER_CASE == the_case_change)
{
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""LOWER CASE");
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() "");
    return new CSpString
(the_string_buffer.toString().toLowerCase());
}
else if(UPPER_CASE ==
the_case_change)
{
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""UPPER CASE");
    CSpValue the_value =
getIMSValue();
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() ");
    return new CSpString
(the_string_buffer.toString().toUpperCase());
}
else
{
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() ");
    return new CSpString
(the_string_buffer.toString());
}
}

CBethIMSFiller.java
iv.

package ImsUtilities;

public class CBethIMSFiller extends CBethIMSDataField
{
    public int getCaseChange()
    {
        return NO_CASE_CHANGE;
    }

    public int getPaddingType()
    {
        return NO_PADDING;
    }
}

CBethIMSPadRight.java
v.

package ImsUtilities;

public class CBethIMSPadRight extends CBethIMSDataField
{

```

```

        public int getCaseChange()
        {
            return NO_CASE_CHANGE;
        }

        public int getPaddingType()
        {
            return PAD_RIGHT;
        }
    }

CBethIMSProcedure.java vi.

package ImsUtilities;

import spider.database.*;
import spider.util.*;
import java.util.*;
import spider.event.*;
import spider.session.*;


    /**
     * An abstract base class for all Precise IMS PAC Data Objects.
     * The NetDynamics PAC technology and specifically the Precise IMS
     * PAC uses a
     * stored procedure interface to IMS transactions on MVS. For the
     * first release
     * of the Precise IMS PAC, only simple <b>string</b> input and output
     * parameters
     * are allowed in the procedure Metadata. Refer to Precise Manual.

For example:

    *
    * <pre>
    * struct outputCE4ZWB01 {
    *     string(12)    outMillOrderNumber;
    *     string(26)    outPurchaseOrderNumber;
    *     string(8)     outProductID;
    *     string(6)     outThickness;
    *     string(8)     outWidth;
    *     string(8)     outOrderedLBS;
    *     string(8)     outAvailableLBS;
    *     string(8)     outUnshippedLBS;
    *     string(8)     outShippedLBS;
    *     string(2)     outPlantCode;
    * }
    *
    * procedure CE4ZWB01 {
    *     text ""Queries Light Flat Roll Mill Orders"";
    *     in string(7)   inShorty;
    *     in string(35)  inPono;
    *     in string(20)  incID;
    *     in string(11)  inProduct;
    *     in string(01)  inType;
    *     in string(5)   inThickFrom;
    *     in string(5)   inThickTo;
    *     in string(7)   inWidthFrom;
    *     in string(7)   inWidthTo;
    *     in string(8)   inDesireDateFrom;
    *     in string(8)   inDesireDateTo;
    *     out struct outputCE4ZWB01[] outTable;
    * }
    * </pre>
    *
    * The IMS transactions, in the case of COBOL programs, have a input
    * and output
    * buffer interface. Single-segment IMS transactions have a single
    * output buffer.


```

```

* Multi-segment IMS transactions have one or more output buffers
(i.e. like
 * multiple rows coming back from SQL or RDBMS stored procedure).
The input and
 * output buffers are bounded character buffers. These buffers
provide the
 * interface to the IMS Message Queue. The schema of the input and
output buffers
 * are specified in the ""01"" level in the COBOL program. The
""01"" level of the
 * COBOL program is key in the development of the Metadata for the
IMS PAC. The
 * Metadata schema must match the ""01"" level in the COBOL exactly.
The simple
 * <b>string</b> restriction on parameters is required to support a
high speed
 * marshalling of data between the NetDynamics PAC service and IMS.
*
 * In addition to the limitation that only simple <b>string</b>
parameter types
 * may be used, there is no PRE or POST processing to the input or
output parameters
 * except to convert to and from EBCDIC and Big/Little Endian (see
Precise Manual
 * for details on the <code>environ</code> column in the <b>appltab
</b> file.
*
 * Currently, there is no automatic padding or truncation of the
input parameters
 * in the IMS PAC service before they are packed into an input buffer
for the
 * IMS Message Queue. Therefore, the out-of-the-box Precise IMS PAC
requires the
 * developer to make sure that the input parameters are formatted
properly.
*
 * This class, and CBethIMSDDataField class, work together to enhance
the features
 * of the Precise IMS PAC to include pre-processing of input
parameters and error
 * handling. All input data fields (parameters) managed by this
class <b>must</b>
 * be derived CBethIMSDDataField classes.
*
 * This class also enforces as Design Pattern which support compile-
time and
 * run-time safe string references. The Design Pattern requires the
developer
 * to include a template in their implementation of the derived class
as follows:
*
* <pre>
* public static final String projectName = "WebOrderStatus";
* public static final String dataObjectName = "imsCE4ZWB01";
*
* public String getProjectName() // implementation of abstract
method
* {
*     return projectName;
* }
*
* public String getDataObjectName() // implementation of abstract
method
* {
*     return dataObjectName;
* }

```

```

*
*   public static CBethIMSProcedure getInstance()
*   {
*       return (CBethIMSProcedure) CSpider.getDataObject(projectName
+ "".""
+ dataObjectName);
*   }
* </pre>
*
* The result is that the developer may reference the name of the
class in a
* high performance and compile-time safe manner with the following:
*
* <code>CSpider.getDataObject(imsCE4ZWB01.dataObjectName);</code>
*
* The project name information support multi-project environments.
Please note
* that only two (2) of the five (5) class members in the template
are enforced
* by the superclass through abstract methods. Java Core Reflection
could be used
* to introspect the class to enforce the rest of the template.
*
* The <code>getInstance()</code> method is helper to get a reference
quickly.
*
* @see CBethIMSDataField
* @version Release: 1.0 Date: 1998/10/20
* @author Matthew J. Stevens, <a
href=""http://www.objectstrategy.com"">ObjectStrategy Inc</a>
*/
public abstract class CBethIMSProcedure extends CSpProcedure
{
    /**
     * Flag used as a condition for debug logging within
the class.
     * Since this member is static final, any decent
compiler should optimize out
     * any static condition. For example:
     *
     * <pre>
     * if(LOCAL_DEBUG) CSpLog.send
(this,CSpLog.USER_DEBUG,""x = "" + x);
     * </pre>
     *
     * is never included in the binary class file if
LOCAL_DEBUG==false.
     */
    public static final boolean LOCAL_DEBUG = true;

    /**
     * Support compile time and run-time safe string
references. Must be
     * implemented by developer in the derived class.
See class description above
     * for the recommended template. Recommended for use
in multi-project applications.
     * @return The name of the project (package) for this
class.
    */
    public abstract String getProjectName();

    /**
     * Support compile time and run-time safe string
references. Must be
     * implemented by developer in the derived class.
See class description above

```

```

        * for the recommended template.
        * @return The name of the class.
        */
    public abstract String getDataObjectName();

    /**
     // Override Spider superclass so that extra Invalid
Parameter Flag is reset
    /**
     public void clearAllValues()
    {
        resetParametersValidFlag();
        super.clearAllValues();
    }

    /**
     // overrided method from CSpFileBased which is the
first method to get called
     // after the File Based object is loaded and init'd.
     //
     // The superclass behaviour is to start the AfterInit
EVENT.  Here we ensure
     // that the superclass behaviour is called first,
then we
    /**
    /**
     * Add behavior to enable certain events
automatically and enforce design pattern.
     * This method should not be called by the developer;
it is called by
     * NetDynamics automatically.  This method is called
after the call to init().
     * This method is implemented to dispatch to
superclass first be added
     * behavior.  Therefore, the AFTER_INIT event will be
dispatched by the
     * superclass before the added behavior occurs.  This
method will force this
     * instance of the class to listen for the
DATA_OBJECT_EVENT's which are
     * automatically handled by the
processDataObjectEvent() method of this class.
     * Please note that this method is called
automatically by NetDynamics is three
     * cases: 1) when the object is loaded from disk
through the intrp model in
     * the CP Worker; 2) when the project graph is cloned
in the CP worker;
     * 3) when the Java client deserializes object in the
browser.
     * @return PROCEED on success or STOP when super
afterInit() fails or design pattern fails.
    */
    public int afterInit()
    {
        //
        // ensure Superclass behaviour called
first
        //
        int command = super.afterInit();
        if(PROCEED != command)
            return command;

        // force the listening of the
DATA_OBJECT_EVENT so that overridden method
        // processDataObjectEvent() is called
    }
}

```

```

always.

        //
        // enableEvents needs to be included
every time there is an afterInit()
        //
        enableEvents
(CSpEvent.DATA_OBJECT_EVENT_MASK);

        //
        // Enforce design pattern that every
CBethIMSProcedure must use derived
        //
        // CBethIMSDataFields for all INPUT
parameters. This forces the developer
        //
        // to specify such in the Properties
of each INPUT parameter.
        //
        // Only need to do this at the server
(helps when using Java Client)
        //
        if(!isReadFromINTRP())
{
        if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""Not in first afterInit() so we
done now"");

        return PROCEED;
}

Enumeration the_fields =
getDataFields().elements();
while(the_fields.hasMoreElements())
{
        Object the_object;
        if(!((the_object =
the_fields.nextElement()) instanceof
CBethIMSDataField))
{
        if
(CSpDataField.PROCEDURE_IN_PARAM ==
((CSpDataField)the_object).getTableNameType())
{
        CSpLog.send(this,CSpLog.ERROR,
""IMS PAC Stored Procedure Input Data Field does not derive from
CBethIMSDataField ["" +
the_object + ""] "");

        return STOP;
}
}
return PROCEED;
}

/***
 * Used internally to automatically handle Data
Object execution events for Comm/Booster condition determination.
 *
 * We are going to capture all CSpDataObjectEvent's
for this Data Object and
 * provide default behaviour for all IMS PAC Data
Objects. It appears that

```

```

        * this method gets called AFTER any derived class
        * <pre>
        * this_onDatabaseErrorEvent()
        * this_onBeforeExecuteEvent()
        * this_onAfterExecuteEvent()
        * </pre>
        * which is built in the Studio. For instance, if
the developer adds these
        * event handlers through the Studio Inspector, they
will be called BEFORE the
        * overridden method below. Therefore, if a
developer wants to customize the
        * the Database Events for a derived
CBethIMSPprocedure, they SHOULD override the
        * method below and dispatch to 'super'. The
developer may use
        * <pre>
        * this_onDatabaseErrorEvent()
        * this_onBeforeExecuteEvent()
        * this_onAfterExecuteEvent()
        * </pre>
        * but they will not get the behavior below first.
        *
        * Please Note: the CSpPage Events for bound Data
Objects are either called
        * before or after the Data Object's Events. For
instance, the
        * <pre>
        * this_onBeforeDataObjectExecuteEvent()
        * </pre>
        * event is called BEFORE the Data Object's events.

However, the
        * <pre>
        * this_onAfterAllDataObjectsExecuteEvent()
        * this_onAfterDataObjectExecuteEvent()
        * this_onDataObjectErrorEvent()
        * </pre>
        * events are all called AFTER the Data Object's
events. Therefore, for
        * application error handling, the developer should
leverage the built-in
        * behaviour of the CBethIMSPprocedure by tapping into
the PAGE's
        * <pre>
        * this_onDataObjectErrorEvent()
        * </pre>
        * which is called AFTER the
CBethIMSPprocedure.processDataObjectEvent() method.
        * In this way, the 'source' Data Object of the Error
Event can be cast to
        * CBethIMSPprocedure and the getCommBoosterCondition
() method can be used to
        * decide what to display to the end user. Example,
if the Comm Booster
        * Condition is COMM_BOOSTER_NO_DATA, then maybe a
friendly message to user
        * explaining why their input led to no returned
data. If the Comm Booster
        * Condition is COMM_BOOSTER_ERROR then maybe the
user should be told to log
        * off and try the application at a later time or
call technical support.
        *
        * @param             event
        * Data Object Event to be processed.
        * @exception         CSpStopException

```

```

* Tells the caller to discontinue processing
* @exception          CSpSkipException
* Tells the caller to skip processing
*/
protected void processDataObjectEvent
(CSpDataObjectEvent event) throws CSpSkipException, CSpStopException
{
    super.processDataObjectEvent(event);
    switch(event.getId())
    {
        //
        // It appears that
ON_DATABASE_ERROR_EVENT is called before
        //
ON_AFTER_EXECUTE_EVENT. In fact, if the result of the
        //
ON_DATABASE_ERROR_EVENT is STOP, then the ON_AFTER_EXECUTE_EVENT
        // is never called.
We will set the proper COMM BOOSTER condition
        // for
ON_DATABASE_ERROR_EVENT.
        //
case
CSpDataObjectEvent.ON_DATABASE_ERROR_EVENT :
{
    if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""ON_DATABASE_ERROR_EVENT"");

CSpDBResultStatus the_result_status = event.getDBResults
().getResultStatus();

CSpLog.send(this,CSpLog.WARNING,""CBethIMSProcedure Result Status =
"" +
the_result_status);
int
the_error_code = the_result_status.getErrorCode();
if(28
== the_error_code)
{

CSpLog.send(this,CSpLog.WARNING,""Precise IMS Service Not
Available"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);
}
else
if(5 == the_error_code)
{

String the_error_message = the_result_status.getErrorMessage();
if( (the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX)) &&
(the_error_message.endsWith(COMM_BOOSTER_RC_83)))
{
CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Type 83
Found"");

setCommBoosterCondition(COMM BOOSTER ERROR);
}
}
}

```

```
}

else if( (the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX))
&&

(the_error_message.endsWith(COMM_BOOSTER_RC_49)) )

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Type 49
Found"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else if( (the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX))
&&

(the_error_message.endsWith(COMM_BOOSTER_RC_5)) )

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Type 5
Found"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else if(the_error_message.equals(COMM_BOOSTER_RC_5_MESSAGE))

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Type 5
Found"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else if(the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX))

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Recieved
But Unknown"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else

{

CSpLog.send(this,CSpLog.WARNING,""No Condition Mapping for Error
Message ("" +
the_error_message + "") "");

setCommBoosterCondition(COMM_BOOSTER_UNKNOWN);

}

}

else
```

```

    {

CSpLog.send(this,CSpLog.WARNING,""No Condition Mapping for Error Code
"" +
the_error_code);

setCommBoosterCondition(COMM_BOOSTER_UNKNOWN);
}

break;
}
case
CSpDataObjectEvent.ON_BEFORE_EXECUTE_EVENT :
{
if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""ON_BEFORE_EXECUTE_EVENT"");
// //
clear the condition before execution
//
resetCommBoosterCondition();

//
//
Upon invalid parameters, the procedure is not executed and
//
the event flow continues (SKIP). The developer may check that
//
the CommBoosterCondition == COMM_BOOSTER_NO_ACTION as well as
//
the Invalid Parameter FLAG. ON_AFTER_EXECUTE_EVENT is never
//
reached.
//

//juan
if(!
areParametersValid())
{

CSpLog.send(this,CSpLog.WARNING,""Procedure has invalid parameters,
will SKIP"");

throw new CSpSkipException("Procedure has invalid parameters");
}

break;
}
case
CSpDataObjectEvent.ON_AFTER_EXECUTE_EVENT :
{
//
//
This is not called if ON_DATABASE_ERROR_EVENT occurs, except
//
(see below)
//
if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""ON_AFTER_EXECUTE_EVENT"");
//
}

```

```

        //
What happens here is that ON_BEFORE_EXECUTE_EVENT resets the
        //
the CommBoosterCondition to NO_ACTION.  checkForNoData() will
        //
set the CommBoosterCondition to SUCCESS or NO_DATA which can be
        //
used by developer
        //
if
(getCommBoosterCondition() == COMM_BOOSTER_NO_ACTION)

//
// Just in case somebody sneaks in more event handling and
// we get to this event from a proceeding ON_AFTER_EXECUTE_EVENT
// or ON_DATABASE_ERROR_EVENT, we don't want to checkForNoData to
// avoid ArrayIndexOutOfBoundsExceptions
//
checkForNoData();

if
(LOCAL_DEBUG)
    CSplLog.send(this,CSplLog.USER_DEBUG,""Comm Booster Condition = "" +
getCommBoosterCondition());

break;
    }
default :
{
    }

CSplLog.send(this,CSplLog.WARNING,""Unknown Data Object Event: "" +
event);
    }
}

/**
 * Used internally to determine the
COMM_BOOSTER_NO_DATA condition.
 * There is a limitation with the IMS PAC caused by
the raw interface of the
 * IMS Message Queue. Precise Comm/Booster cannot
distinguish between an
 * IMS transactions which really abends and one which
returns NO DATA.
 * If an IMS transaction (with transaction property
RESPONSE_MODE) doesn't
 * insert a properly sized buffer to the I/O PCB then
Comm/Booster considers
 * this* an abend. Therefore, in order to propagate a
NO DATA conditon,
 * <b>All IMS Transactions used with Precise PAC and
this class</b> must
 * return a single record with all SPACES.
 */
protected void checkForNoData()
{

```

```

        //
        // The Multi-Segment stored
procedures for Precise IMS return a single
        // CSpDBProcedureResult which
encapsulates a CSpDBProcedureParam which
        // encapsulates a vector of
TNdValue's. When multiple rows are returned,
        // there will still only be one (1)
TNdValue for each column; that is, a
        // procedure which returns eight (8)
columns of data will have eight (8)
        // TNdValue's in the vector
regardless of the number of rows. The
        // TNdValue's are surely
""constructed types"". There is an interface to
        // extract both TNdValue and CSpValue
Java types from the CSpDBProcedure-
        // Param. We will stick with the
CSpValue since TNdValue is more advanced.
        //
        CSpDBProcedureResult the_result =
(CSpDBProcedureResult) getLastResults();
        CSpDBProcedureParam
the_result_parameters = the_result.getReturnParameters();
        int the_parameter_count =
the_result_parameters.getNumParameters();
        if(LOCAL_DEBUG)
            CSpLog.send
(this,CSpLog.USER_DEBUG,""The Number of Parameters = "" +
the_parameter_count);

        try
{
        CSpVector the_vector
= (CSpVector) the_result_parameters.getParameter(0);
        if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""The first vector size = "" +
the_vector.size());
        if(the_vector.size()
< 1)
{
        //
        //
should never be reached
        //

CSpLog.send(this,CSpLog.ERROR,
""IMS Transaction Design Pattern Failure - Successful Execution
always return at least one row"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);
        }
        else if
(the_vector.size() > 1)
{
        if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,
""More than one (1) row returned, probably good data"");

```



```

        */
        public static final String COMM_BOOSTER_RC_49 = ""
49"";
        /**
         * String for matching error conditions in
Comm/Booster.
        */
        public static final String COMM_BOOSTER_RC_5 = ""
5"";
        /**
         * String for matching error conditions in
Comm/Booster.
        */
        public static final String COMM_BOOSTER_RC_5_MESSAGE
= ""Comm/Booster rc = 5"";

        /**
         * Unknown Condition for Comm/Booster.
        */
        public static final int COMM_BOOSTER_UNKNOWN = -1;
        /**
         * Initial condition for Comm/Booster before
execution.
        */
        public static final int COMM_BOOSTER_NO_ACTION = 0;
        /**
         * The default condition for Comm/Booster when no
error is found after execution.
        */
        public static final int COMM_BOOSTER_SUCCESS = 1;
        /**
         * Value representing a system failure in the IMS PAC
system. This includes
         * MVS connectivity problems, abend of Precise TCP/IP
relay on MVS, major
         * error in IMS PAC or Comm/Booster middleware
binaries.
        */
        public static final int COMM_BOOSTER_ERROR = 2;
        /**
         * Value representing a condition where an IMS
Transaction was successfully executed but no data was returned.
        */
        public static final int COMM_BOOSTER_NO_DATA = 3;

        /**
         * State of Comm/Booster.
        */
        transient protected int _CommBoosterCondition =
COMM_BOOSTER_NO_ACTION;
        /**
         * State of the input parameters after customized
validation and pre-processing.
        */
        transient protected boolean _ValidProcedureParameters
= true;

        //
// Comm Booster Condition may be accessed publicly
//
/**
     * Access the condition of the Data Object with
respect to Comm/Booster behavior.
     * Most useful after Data Object Execution.
     * @return COMM_BOOSTER_NO_ACTION,
COMM_BOOSTER_SUCCESS, COMM_BOOSTER_ERROR, COMM_BOOSTER_NO_DATA

```

```

        */
    public int getCommBoosterCondition()
    {
        return _CommBoosterCondition;
    }

    /**
     * Checks the state of the input parameters.
     * @return true if any individual input parameter
vetoed the Data Object execution
     */
    public boolean areParametersValid()
    {
        return _ValidProcedureParameters;
    }

    /**
     * Used internally to reset the input parameter
validity flag.
     */
    protected void resetParametersValidFlag()
    {
        _ValidProcedureParameters = true;
    }

    /**
     * Used internally to vote NO to Data Object
execution.
     */
    protected void invalidateParameters()
    {
        _ValidProcedureParameters = false;
    }

    /**
     * Used internally to reset the Comm/Booster
condition before Data Object execution.
     */
    protected void resetCommBoosterCondition()
    {
        _CommBoosterCondition =
COMM_BOOSTER_NO_ACTION;
    }

    /**
     * Used internally to set the Comm/Booster condition
during or after Data Object execution.
     */
    protected void setCommBoosterCondition(int
pCondition)
    {
        if( (pCondition !=
COMM_BOOSTER_SUCCESS) &&
(pCondition !=
COMM_BOOSTER_ERROR) &&
(pCondition !=
COMM_BOOSTER_UNKNOWN) &&
(pCondition !=
COMM_BOOSTER_NO_DATA) )
        {
            CSpLog.send
(this,CSpLog.ERROR,""Unknown Comm Booster Condition -> "" +
pCondition);
            _CommBoosterCondition
= COMM_BOOSTER_UNKNOWN;
        }
    }
}

```

```

        }
        _CommBoosterCondition = pCondition;
        if(LOCAL_DEBUG)
        {
            String the_message =
            ""Comm Booster Condition of CBethIMSProcedure is "";
            switch(pCondition)
            {
                case
                COMM_BOOSTER_SUCCESS :
                {
                    the_message = the_message + ""COMM_BOOSTER_SUCCESS"";

                    break;
                }
                case
                COMM_BOOSTER_ERROR :
                {
                    the_message = the_message + ""COMM_BOOSTER_ERROR"";

                    break;
                }
                case
                COMM_BOOSTER_NO_DATA :
                {
                    the_message = the_message + ""COMM_BOOSTER_NO_DATA"";

                    break;
                }
                case
                COMM_BOOSTER_UNKNOWN :
                {
                    the_message = the_message + ""COMM_BOOSTER_UNKNOWN"";

                    break;
                }
            }
            CSpLog.send
            (this,CSpLog.USER_DEBUG,the_message);
        }
    }
}

```

vii.

```

Credit.java
package CreditApp;

import java.io.*;
import java.text.*;
import spider.*;
import spider.database.*;
import java.util.*;
import spider.visual.*;
import spider.session.*;
import spider.util.*;

//[[SPIDER_CLASS BEGIN
public class Credit implements Serializable
{
    private String pastCust = new String();
    private String pastYear = new String();

```

```
private String pastName = new String();
private String plate = new String();
private String semiFinished = new String();
private String pipe = new String();
private String sheet = new String();
private String rail = new String();
private String reclamation = new String();
private String other = new String();
private String tinPlate = new String();
private String bar = new String();
private String secondary = new String();
private String otherSpec = new String();
private String businessDivision = new String();
private String domestic = new String();
private String custCode = new String();
private String district = new String();
private String product = new String();
private String cpnyName = new String();
private String incorporatedName = new String();
private String cpnyDivisionName = new String();
private String pOName = new String();
private String headquarterAddress = new String();
private String headquarterBoxNum = new String();
private String headquarterCity = new String();
private String headquarterZip = new String();
private String headquarterForeignAddress = new String();
private String headquarterPhone = new String();
private String headquarterFax = new String();
private String headquarterWeb = new String();
private String purchaseThroughBranch = new String();
private String branchName = new String();
private String branchDivision = new String();
private String branchStAddress = new String();
private String branchCity = new String();
private String branchZip = new String();
private String branchForeignAddress = new String();
private String branchPhone = new String();
private String branchFax = new String();
private String invoiceCorrespondence = new String();
private String otherStreet = new String();
private String otherBoxNum = new String();
private String otherCity = new String();
private String otherZip = new String();
private String contactWithBeth = new String();
private String contactBethName = new String();
private String headquarterState = new String();
private String invoiceForeignAddress = new String();
private String branchState = new String();
private String otherState = new String();
private String apSystem = new String();
private String apMailedTo = new String();
private String buyersName = new String();
private String buyersPhoneNum = new String();
private String buyersEmail = new String();
private String apSupervisorName = new String();
private String apPhoneNum = new String();
private String apEmail = new String();
private String financialSupportName = new String();
private String dun1 = new String();
private String dun2 = new String();
private String dun3 = new String();
private String fedID1 = new String();
private String fedID2 = new String();
private String bankName = new String();
private String bankStreet = new String();
private String bankCity = new String();
```

```
<tr><td colspan=""10"" align=left>City  
&nbsp;<b>-Required</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;**BankCity**</td></tr>  
  
<tr>  
<td colspan=""10"" align=left>State  
&nbsp;<b>-Required</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
**BankState**  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Other&nbsp;&nbsp;**BankOther**</td></tr>  
<tr>  
<td colspan=""10"" align=left>Zip&nbsp;<b>-Required</b>&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;**BankZip**</td></tr>  
<tr>  
<td colspan=""10"" align=left>Account officer's name&nbsp;&nbsp;  
**AcctOfficerName**</td></tr>  
  
<tr>  
<td colspan=""10"" align=left>Phone Number&nbsp;&nbsp;  
**AcctOfficerPhoneNum**</td></tr>  
  
<tr>  
<td colspan=""10"" align=left>&nbsp;</td></tr>  
<tr>  
<td colspan=10>&nbsp;</td></tr>  
  
<tr>  
<td colspan=""10"" align=left><b>Note:</b><i>In today's business  
environment, banks are sometimes reluctant to respond  
to a prospective creditor's request, whether due to individual bank  
policy or time/manpower  
constraints. This significantly hampers our ability to complete a  
credit check. To expedite this  
process, we ask that you contact your bank and request them to fax  
the bank's experience/relationship  
to Bethlehem Steel Corporation at (610) 694-3287 or email to:  
**EmailBankInfo**</td></tr>  
</table>  
</td>  
  
<table border=0 cellpadding=2 cellspacing=2 width=650>  
<tr>  
<td colspan=10>&nbsp;</td></tr>  
<tr>  
  
<td colspan=10 align=right>**GoToApp2** &nbsp; | &nbsp;**GoToApp4**  
</td></tr>  
<tr>  
<td colspan=10 align=left>**GoToCreditHmPg**</td></tr></table>  
  
**SPIDERSESSION**  
[[/SPIDERPAGE]]
```

v.

```

CreditApp4.htm
<HTML>
<HEAD>
<TITLE>Credit Application</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditApp4]]

<center>
<table border=0 cellspacing=0 cellpadding=0 width=650>
<tr>
<th colspan=10 align=left bgcolor="#666699"><font size=4
color="white">&nbsp;Trade References</font></th></tr>
<tr>
<td colspan=10 align=left>Preferably other large volume suppliers.
&nbsp;<b>-Name and City/St/Zip are Required</b></td></tr>

<tr>
<td colspan=2 align=right>(1)&nbsp;&nbsp;Name:</td>
<td colspan=8 align=left>**TradeRefName1**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;Street:</td>
<td colspan=8 align=left>**TradeRefStreet1**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;City/St/Zip:</td>
<td colspan=8 align=left>**TradeRefCityStZip1**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Attn:**TradeRefAttn1**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Phone:**TradeRefPhone1**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Fax:**TradeRefFax1**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=right>(2)&nbsp;&nbsp;Name:</td>
<td colspan=8 align=left>**TradeRefName2**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;Street:</td>
<td colspan=8 align=left>**TradeRefStreet2**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;City/St/Zip:</td>
<td colspan=8 align=left>**TradeRefCityStZip2**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Attn:**TradeRefAttn2**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Phone:**TradeRefPhone2**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Fax:**TradeRefFax2**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=right>(3)&nbsp;&nbsp;Name:</td>
<td colspan=8 align=left>**TradeRefName3**</td></tr>
<tr>

```

```

<td colspan=2 align=right>&nbsp;&nbsp;Street</td>
<td colspan=8 align=left>**TradeRefStreet3**</td></tr>
<tr>
<td colspan=2 align=right>&nbsp;&nbsp;City/St/Zip:</td>
<td colspan=8 align=left>**TradeRefCityStZip3**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Attn:**TradeRefAttn3**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Phone:**TradeRefPhone3**</td></tr>
<tr>
<td colspan=2 align=left>&nbsp;</td>
<td colspan=8 align=left>Fax:**TradeRefFax3**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10><hr color=""black"" width=650></hr></td></tr>

<tr>
<td colspan=10 align=left><b>Thank you for your time in completing
this application. Click
the ""Submit"" button below and we will get to work on processing
your application as quickly
as possible. We look forward to doing business with you!</b></td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=center>**Submit**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;</td>
</td></tr></table>

<table border=0 cellpadding=2 cellspacing=2 width=650>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=right>**GoToApp3**</td></tr>
<tr>
<td colspan=10 align=left>**GoToCreditHmPg**</td></tr>
</TABLE>

```

From
 SPIDERSESSION
 [[/SPIDERPAGE]]

```

</BODY>
</HTML>

```

vi.

```

CreditApplication.htm
<HTML>
<HEAD>
<TITLE>Credit Application</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditApplication]]
<center>

<TABLE BORDER=0 CELLSPACING=0 CELL_PADDING=0 WIDTH=650>
<tr>

```

```

<td colspan=""10"" align=center>**logobeth**
</td></tr>
</TABLE>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<th colspan=""10"" align=center bgcolor="#666699"><font size=4
color="white">Credit Application</font></th>
</tr>

<tr>
<td colspan=""10"" align=center>&ampnbsp</td></tr>
<td colspan=""10"" align=center><i>(Note: We retain all information
in strict confidence)</i></td></tr>

<tr>
<td colspan=""10"" align=center>&ampnbsp</td></tr>
<td colspan=""10"" align=left>We want to sell to you! We do not want
"Credit" to be a reason to delay,
refuse, or lose your order. By filling out all areas of our on-line
credit application, we will be
able to handle the credit process in the fastest possible manner. We
encourage electronic submission,
however, you may use the print option and fax the completed form to:
(610) 694-3287</td></tr>

<tr>
<td colspan=""10""><hr width=""650"" color=""#666699""></td></tr>
<tr>
<th colspan=""10"" align=left><b>Note: If your initial purchase is
intended to be via <i>Metalsite</i>, please
fill out the credit application form through the 'Buyer
Registration' process at **MetalSiteLink**</td></tr>
<tr>
<td colspan=""10""><hr width=""650"" color=""#666699""></td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=""650"">
<tr>
<td colspan=""10"">&ampnbsp</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>Have you ever purchased
from Bethlehem in the past?
&ampnbsp&ampnbsp&ampnbsp&ampnbsp**PastCust**</td></tr>

<td colspan=""10"">&ampnbsp</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>If yes, when and under
what name?
&ampnbsp&ampnbsp&ampnbspYear- **PastYear**&ampnbsp&ampnbsp&nbsp;Name- **PastName**
</td></tr>

<tr>
<td colspan=""10"" width=""650"">&ampnbsp</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>What product or products
do you plan to purchase initially?</td></tr>

<tr>
<td colspan=""2"" width=""190"" align=left>**Plate**Plate</td>
<td colspan=""2"" width=""200"" align=left>
**SemiFinished**SemiFinished</td>
<td colspan=""2"" width=""190"" align=left>**Pipe**Pipe</td>

```

```

<td colspan=""4"" width=""80"">&nbsp;</td></tr>

<tr>
<td colspan=""2"" width=""190"" align=left>**Sheet**Sheet</td>
<td colspan=""2"" width=""200"" align=left>**Rail**Rail</td>
<td colspan=""2"" width=""190"" align=left>**Reclamation**Reclamation
</td>
<td colspan=""4"" width=""80"">&nbsp;</td></tr>

<tr>
<td colspan=""2"" width=""190"" align=left>**TinPlate**Tin Plate</td>
<td colspan=""2"" width=""200"" align=left>**Bar**Bar</td>
<td colspan=""2"" width=""190"" align=left>**Secondary**Secondary
</td>
<td colspan=""4"" width=""80"">&nbsp;</td></tr>

<tr>
<td colspan=""10"" width=""650"">**Other**Other &nbsp; Specify:&nbsp;
&nbsp;**OtherSpec**</td>

<tr>
<td colspan=""10"" width=""650"">&nbsp;</td></tr>

<tr>
<td colspan=""10"" width=""650"" align=left>Do you know which
Bethlehem business division will ship your order?</td></tr>

<tr>
<td colspan=""1"" width=""25"">&nbsp;</td>
<td colspan=""9"" width=""625"" align=left>**BusinessDivision**</td>
</tr>

<tr>
<td colspan=""10"" width=""650"">&nbsp;</td></tr>

<tr>
<td colspan=""10"" width=""650"">Is your order domestic or export?
&nbsp;&nbsp;&nbsp;**Domestic**</td></tr>
</table>






```

SPIDERSESSION
[[/SPIDERPAGE]]

```

</BODY>
</HTML>

```

vii.

```

CreditDeptHomePg.htm
<HTML>
<HEAD>
<TITLE>CreditDept Home Page</TITLE>
</HEAD>

```

```

[[SPIDERPAGE NAME=CreditDeptHomePg]]

<center>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr><td colspan=10 align=center>**logobeth**</td></tr>
<tr><th colspan=10 align=center><font size=6>Credit Information
</font></th></tr>
<tr><td colspan=10 align=center bgcolor="#666699">&ampnbsp</td></tr>
<tr><td colspan=10 align=center>&ampnbsp</td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=500>
<tr><td align=center><font size=5><ul><li>**CreditInquiryPg**</li>
</ul></font></td></tr>
<tr><td align=left>-<i> A service for our customers whereby they can
provide access
by others to their credit history with Bethlehem.</i></td></tr>
</TABLE>

<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr><td colspan=10 align=center><hr width=""650"" color=""black"">
</td></tr>
<tr><th colspan=10 align=left><font size=4>New Customers</font></th>
</tr>
<tr>
<td colspan=10 align=left>
<ul>
<li>**CreditAppPgNewCust**</li>
<br>
<li>**CreditRepPgNewCust**</li>
</ul></td></tr>
<tr>
<td colspan=10 align=center><hr width=""650"" color=""black""></td>
</tr>
<tr>
<td colspan=10 align=left>
<tr>
<th colspan=10 align=left><font size=4>Existing Customers </th></tr>
<tr>
<td colspan=10 align=left>
<ul>
<li>**WebOrderStatusLink** &nbsp;&nbsp;&nbsp; (I.D. & Password for
'<i>Customer
Corner' required)</i></li>
<br>
<li>**CreditInstructions**</li>
</ul></td></tr>
<tr>
<td colspan=10 align=center><hr width=""650"" color=""black""></td>
</tr>
<tr>
<td colspan=10 align=left>
<tr>
<th colspan=10 align=left><font size=4>Bethlehem Steel Policies</th>
</tr>
<tr>
<td colspan=10>
<ul>
<li>**BethCreditPolicy**</li>
<br>
<li>**BethDiscountPolicy**</li></ul></td></tr>
</TABLE>

<BR>***HomePage**

```

```

**SPIDERSESSION**
[ [/SPIDERPAGE]] viii.

</BODY>
</HTML>

CreditDiscountPolicy.htm
<HTML>
<HEAD>
<TITLE>Credit Discount Policy</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditDiscountPolicy]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
<tr>
<td colspan=10 align=center>&ampnbsp</td></tr>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color="white">Bethlehem's Standard Terms & Cash Discount Policy
</font></th></tr>
<tr>
<td colspan=10 align=left>&ampnbsp</td></tr>
<tr>
<th colspan=10 align=left>Bethlehem Steel Corporation's standard,
unsecured / 'open', terms for
most products are<br>30-3/4-10. There are other unsecured terms that
are determined by product and
method of shipment.</th></tr>
<tr>
<td colspan=10 align=center>&ampnbsp</td></tr>
<tr>
<th colspan=10 align=left>Payment, regardless of the specific terms,
is based on <u>receipt</u>
of your remittance. For example, on standard terms of 30-3/4-10, we
will consider a payment
""prompt"" if the monies are received on or within 30 days from the
invoice date. Likewise, in
order to take advantage of the cash discount offered, payment must be
received within the period
specified in the terms. In the example above, receipt is required
within the 10 days of the
invoice date.</th></tr>
<tr>
<td colspan=10 align=center>&ampnbsp</td></tr>
<tr>
<th colspan=10 align=left>For the convenience of our customers,
Bethlehem offers the option to
group invoices for payment (on the 'received' basis) twice per month.
</th></tr>
<tr>
<th colspan=10 align=left>Under these guidelines:</th></tr>
<tr>
<td colspan=10 align=left>&ampnbsp</td></tr>

<tr>
<th colspan=10 align=left>
<dl>
<dt>On a Discount Basis -</dt>
<dd>Invoices dated from the 1st to the 15th - Payment due by the
25th of the same month.</dd>
<dd>Invoices dated from the 16th to the 31st - Payment due by the
10th of the following month.</dd></dl></th></tr>

```

```

<tr>
<th colspan=10 align=left>
<dl>
<dt>On a Net or ""Prompt"" Basis -</dt>
<dd>Invoices dated from the 1st to the 15th - Payment due by the
10th of the following month.</dd>
<dd>Invoices dated from the 16th to the 31st - Payment due by the
25th of the following month.</dd></dl></th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center><font size=3 color="#666699"><b>
QUESTIONS? email:</b>
<a href=""mailto:bscredit@bethsteel.com"">bscredit@bethsteel.com</a>
</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>**BackToCreditHomePage**</td></tr>

</TABLE>

```

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

ix.

CreditInquiry.htm

```

<HTML>
<HEAD>
<TITLE>Credit Inquiry</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditInquiry]]
<CENTER>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr></table>
<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<TH ALIGN=center VALIGN=MIDDLE bgcolor="#666699"><FONT SIZE=4
color=""white""><B>Credit Inquiry</B></FONT></TH>
</TABLE>

<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<TH ALIGN=left VALIGN=left><FONT SIZE=3>As a matter of policy,
Bethlehem shares trade experience as a service to its customers.
</FONT></TH>
</TABLE>

<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>
<TH ALIGN=left VALIGN=left><FONT SIZE=3>This information is available
in an easy to use electronic format and can only
be accessed by obtaining a 6-digit code and 4-digit ""CIN"" number
from individual customers of Bethlehem.</FONT></TH>
</TABLE>

<BR>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=60%>

```

<TH ALIGN=left VALIGN=left>YOU, as a third party inquiring of Bethlehem's experience with one of our customers, will be allowed access to certain credit information. In order to access this information, you will need to enter the following information below:</TH>

</TABLE>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=50%>
<tr>
<TH ALIGN=left VALIGN=left>

Your company name/ address/ phone #

Your company's DUNS # and Fed I.D. #

Your name/ title

A six (6) digit code # supplied to you by our customer

A four (4) digit ""CIN"" (commercial identification number)
supplied to you by our customer.
</TH></tr>
<tr>
<td colspan=""10""> </td></tr>
<tr>
<td colspan=""10"" align=center>**Continue**</td></tr>
<tr>
<td colspan=""10""> </td></tr>
<tr>
<td colspan=""10"" align=center>**BackToCreditHomePage**</td></tr>
</TABLE>

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

x.

CreditInquiryForm.htm

<HTML>
<HEAD>
<TITLE>CreditInquiry Form</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditInquiryForm]]
<center>
<TABLE BORDER=0 CELLSPACING=1 CELLPADDING=1 WIDTH=650>
<tr>
<td colspan=""10"" align=""center"">**logobeth**</td></tr>
<tr>
<td colspan=""10"" align=""center"">
ErrorMessage</td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=1 CELLPADDING=1 WIDTH=650>
<TH ALIGN=LEFT COLSPAN=""10"" bgcolor=""#666699""><FONT SIZE=3
color=""white""> Enter your company name, address and phone
number:</TH>
<TR>
<TD COLSPAN=""2""> </TD>
<TD COLSPAN=""3"">Company Name</TD>
<TD COLSPAN=""1""> </TD>
<TD colspan=""4"">**CpnyName**</TD></TR>
<TR>

86

```

<TD colspan=""2"">&nbsp;</td>
<TD align=left colspan=""3"">Street</td>
<TD colspan=""1"">&nbsp;</td>
<TD colspan=""4"">**CpnyStreet**</TD></TR>

<TR>
<TD colspan=""2"">&nbsp;</td>
<TD colspan=""3"" align=left>City </td>
<TD colspan=""1"">&nbsp;</td>
<TD colspan=""4"">**CpnyCity**</td></tr>

<tr>
<td colspan=""2"">&nbsp;</td>
<td colspan=""3"" align=left>State</td>
<td colspan=""1"">&nbsp;</td>
<td colspan=""4"">**CpnyState** &nbsp;&nbsp;Other&nbsp; **CpnyOther**</td></tr>

<tr>
<td colspan=""2"">&nbsp;</td>
<td colspan=""3"" align=left>Country</td>
<td colspan=""1"">&nbsp;</td>
<td colspan=""4"">**CpnyCountry** &nbsp;&nbsp;Zip Code&nbsp;**CpnyZip**</td></tr>

<TR>
<TD colspan=""2"">&nbsp;</td>
<TD align=left colspan=""3"">Phone Number</td>
<TD colspan=""1"">&nbsp;</td>
<TD colspan=""4"">**CpnyPhone**</td></tr>

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<TH ALIGN=LEFT COLSPAN=""10"" bgcolor="#666699"><FONT SIZE=3 color="white">&nbsp;Enter your company's DUN'S # and Federal I.D.#:</font></TH></tr>
<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">DUNS #</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**Dn1**-**Dn2**-**Dn3**</td></tr>

<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"" align=left>Fed I.D.#</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**FdID1**-**FdID2**</td></tr>

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<TH ALIGN=LEFT COLSPAN=""10"" bgcolor="#666699"><FONT SIZE=3 color="white">&nbsp;Enter your name and title:</font></TH></tr>
<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">Name</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**CustName**</td></tr>

<TR>
<TD COLSPAN=""2"">&nbsp;</TD>
<TD COLSPAN=""3"">Title</TD>
<TD COLSPAN=""1"">&nbsp;</TD>
<TD colspan=""4"">**CustTitle**</td></tr>

```

```

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<th align=LEFT COLSPAN=""10"" bgcolor="#666699"><font size=3
color=""white"">&nbsp;Enter the six (6) digit code # supplied by our
customer (subject company) to you:</font></th></tr>
<tr>
<td colspan=""2"">&nbsp;</td>
<td colspan=""3"">Code #</td>
<td colspan=""1"">&nbsp;</td>
<td colspan=""4"">**CstCde**</td></tr>

<tr>
<td colspan=""10"">&nbsp;</td></tr>
<tr>
<th align=LEFT COLSPAN=""10"" bgcolor="#666699"><font size=3
color=""white"">&nbsp;Enter the four (4) digit ""CIN"" number
supplied by our customer (subject company) to you:</font></th></tr>
<tr>
<td colspan=""2"">&nbsp;</td>
<td colspan=""3"">CIN #</td>
<td colspan=""1"">&nbsp;</td>
<td colspan=""4"">**CINNum**</td></tr>
</table>

<br>
<table border=0 cellspacing=1 cellpadding=1 width=550 align=center>
<tr>
<td align=left colspan="10"><font size=2>By clicking the button
below, you represent that the information retrieved will be
used only by your company, with no liability to Bethlehem Steel
Corporation, for
the sole purpose of assisting in <u>your</u> credit evaluation of
the subject company. In
addition, you represent that the subject company has specifically
given to you, or
your company, 'Bethlehem Steel Corporation' as a trade reference
for the purpose
of obtaining normal trade credit information.</font></td>
</tr></table>
<tr>
<td colspan=""10"" align=center>&nbsp;</td></tr>
<tr>
<td colspan=""3"">&nbsp;</td>
<td colspan=""2"" align=center>**Agree**</td>
<td colspan=""2"" align=center>**Decline**</td>
<td colspan=""3"">&nbsp;</td>
</tr>
<tr>
<td colspan=""10"" align=center>&nbsp;</td></tr>
<tr>
<td colspan=""10"" align=center>**BackToCreditHomePage**</td></tr>
</table>

```

```

<br>
<br>
<br>
<br>
**SPIDERSESSION**
[[/SPIDERPAGE]]

```

```

</body>
</html>

```

```
CreditInquiryResults.htm
<HTML>
<HEAD>
<TITLE>Credit Inquiry Results</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditInquiryResults]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
</table>

<br><br>

<TABLE BORDER=""0"" CELLSPACING=""0"" CELLPADDING=""0"" WIDTH=650>
<tr>
<td colspan=10 align=left bgcolor="#666699"><font size=4
color="white">&nbsp;Your inquiry on : **CustName**&nbsp;
**CustLocation**</td></tr>
<tr>
<td colspan=10 align=left bgcolor="#666699"><font size=4
color="white">
&nbsp;has produced the following history:</font></td></tr></table>

<TABLE BORDER=""0"" CELLSPACING=""0"" CELLPADDING=""0"" WIDTH=650>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=250>Years Sold&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;</th>
<td colspan=4 align=left width=250>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**YrsSold**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=250>High Credit&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;</th>
<td colspan=4 align=left width=250>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**HighCreidit**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=250>Terms&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;</th>
<td colspan=4 align=left width=250>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**Terms**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=300>Payment Days&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;</th>
<td colspan=4 align=left width=350>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**PaymentDays**</td></tr>
```

```

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=2 align=left width=150>&nbsp;</td>
<th colspan=4 align=right width=300>Last Sale&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;</th>
<td colspan=4 align=left width=350>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;**LastSale**</td></tr>

</TABLE>

<table border=0 cellpadding=2 cellspacing=2 width=650>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=5 align=left>**BackToCreditHomePage**</td>
<td colspan=5 align=right>**BackToInquiryForm**</td></tr>
</table>

<BR>
<BR>
<BR>
**SPIDERSESSION**
[[/SPIDERPAGE]]

</BODY>
</HTML>

```

xii.

```

CreditPolicy.htm
<HTML>
<HEAD>
<TITLE>Credit Policy</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditPolicy]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELL_PADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>

<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color="white">Bethlehem Steel Corporation Credit Policy</font></th>
</tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left>Bethlehem's Finance Department (Credit
Division) is responsible for
determining what constitutes an acceptable credit risk to Bethlehem
and for protecting its investment in Accounts Receivable consistent
with
Bethlehem's vision, strategies and objectives. It will develop
reasonable

```

and prudent credit policies together with the Business Units and Field Sales to aid in maximizing consolidated Return on Net Assets. Credit exposure will be managed to minimize days sales outstanding (DSO) and bad debts.</th></tr>

```

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>**BackToCreditHomePage**</td></tr>

</TABLE>

```

SPIDERSESSION
[[/SPIDERPAGE]]

</BODY>
</HTML>

xiii.

CreditRepNewCust.htm
<HTML>
<HEAD>
<TITLE>Credit Representative</TITLE>
</HEAD>

[[SPIDERPAGE NAME=CreditRepNewCust]]

```

<center>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color="white">Contact Your Credit Representative</font></th></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left><u><b>We believe that communication is the
key to great customer
service.</u></b> Here is a tool to help you reach the credit person
within our
organization that handles your account.</td></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left>Bethlehem's Credit Division is a
decentralized function with credit
professionals located at each of our business divisions and at
corporate.</td></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left>Generally speaking, your company's
geographic location coupled with
the location of Bethlehem's business division that will produce and
ship your order, are the
criteria that determine the representative handling your account.
(If purchasing product
from more than one plant location, then the 'rule of thumb' defaults
to the representative
of the plant shipping the greater tonnage.)</td></tr>

```

```

<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left><hr width=""100%"" color=""black""></hr>
</td></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left><font size=4>New Customer</font></th></tr>
<tr>
<td colspan=10 align=left>If you are a prospective or new customer -
<b>CHOOSE</b> the geographic
area nearest you, product, and the Bethlehem <u>business division</u>
to produce your order:</td></tr>
<tr>
<td colspan=10 width=""650"">&nbsp;</td></tr>
<tr>
<td colspan=2 align=left><b>District</b>&nbsp;**District**&nbsp;
&nbsp;&nbsp;</td>
<td colspan=2 align=left><b>Product</b>&nbsp;</td>
<td colspan=6 align=left>**Product**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=left><b>Business Division</b>
&nbsp;&nbsp;**BusDivision**</td></tr>

<tr>
<td colspan=10 align=left>&nbsp;</td></tr>

<tr>
<td colspan=10 align=center>**CreditRepNewCustBttn**</td></tr>
<tr>
<td colspan=10 align=center>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center>**GoToCreditHmPg**</td></tr>
</TABLE>

```

SPIDERSESSION
 [[/SPIDERPAGE]]

</BODY>
 </HTML>

xiv.

CreditRepNewResults.htm
 <HTML>
 <HEAD>
 <TITLE>Credit Representative Results</TITLE>
 </HEAD>
 [[SPIDERPAGE NAME=CreditRepNewResults]]
 <center>
 <TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
 <tr>
 <td colspan=10 align=center>**logobeth**</td></tr>
 </TABLE>

```

<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color="white">Your Credit Representative based on the information
entered is:</th></tr>
<tr><td colspan=10 align=left>&nbsp;</td></tr>
<tr><th colspan=10 align=center>**RepName**</th></tr>
<tr><th colspan=10 align=center>**RepTitle**</th></tr>
<tr><td colspan=10 align=center>&nbsp;</td></tr>
<tr><th colspan=10 align=center>Phone- **RepPhone**</th></tr>
<tr><th colspan=10 align=center>Fax- **RepFax**</th></tr>
<tr><td colspan=10 align=center>&nbsp;</td></tr>
<tr><td colspan=10 align=center><font size=2>To communicate via
email, click here:</font></td></tr>
<tr><th colspan=10 align=center><a href=""mailto:$$$$$$"><font
color="blue">**EmailCreditRep**</font></a></th></tr>
<tr><td colspan=10 align=center>&nbsp;</td></tr>
<tr><td colspan=5 align=left>**GoToCreditHmPg**</td>

<td colspan=5 align=right>**GoToCreditRepNewCust**</td></tr>
</table>

```

```

<BR>
**SPIDERSESSION**
[[/SPIDERPAGE]]

```

```

</BODY>
</HTML>

```

xv.

```

InquiryTrackScreen.htm
<HTML>
<HEAD>
<TITLE>Credit Inquiry Tracking Screen</TITLE>
</HEAD>

[[SPIDERPAGE NAME=InquiryTrackSceen]]
<center>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>
<BR>
<tr>
<td align=center width=100%><font size=4>Credit Inquiry Tracking
Admin Screen</font></td></tr>
</table>

<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=2 WIDTH=100%>
<tr>
<td align=center width=7% bgcolor="#666699"><font size=2
color="white">Customer Code</font></td>
<td align=center width=15% bgcolor="#666699"><font size=2
color="white">Company Name</font></td>
<td align=center width=20% bgcolor="#666699"><font size=2
color="white">User Name & Title</font></td>
<td align=center width=10% bgcolor="#666699"><font size=2
color="white">Phone Number</font></td>
<td align=center width=20% bgcolor="#666699"><font size=2
color="white">
```

```

color=""white"">Address</font></td>
<td align=center width=4% bgcolor="#666699"><font size=2
color=""white"">CIN Number</font></td>
<td align=center width=8% bgcolor="#666699"><font size=2
color=""white"">Duns Number</font></td>
<td align=center width=8% bgcolor="#666699"><font size=2
color=""white"">Fed ID</font></td>
<td align=center width=8% bgcolor="#666699"><font size=2
color=""white"">Time Stamp</font></td></tr>

<BR><!-- [[SPIDERREPEAT NAME=Repeated1]] -->
<tr>
<td align=center width=7%><font size=2>**CusCode**</font></td>
<td align=center width=15%><font size=2>**CpnyNam**</font></td>
<td align=center width=20%><font size=2>**Nam**&nbsp;&nbsp;&nbsp;
**Tl**</font></td>
<td align=center width=10%><font size=2>**Ph**</font></td>
<td align=center width=20%><font size=2>**Str**&nbsp;**City**&nbsp;
**State**&nbsp;**Zip**&nbsp;**Country**&nbsp;**Other**</font></td>
<td align=center width=4%><font size=2>**CinNumber**</font></td>
<td align=center width=8%><font size=2>**Dn**</font></td>
<td align=center width=8%><font size=2>**Fd**</font></td>
<td align=center width=8%><font size=2>**TimeSt**</font></td>
</tr>

<!-- [[/SPIDERREPEAT]] -->
</table>

```

```

**SPIDERSESSION**
[[/SPIDERPAGE]]
</BODY>
</HTML>

```

xvi.

```

Override.htm
<HTML>
<HEAD>
<TITLE>OverrideTest</TITLE>
</HEAD>

[[SPIDERPAGE NAME=Override]]

<center>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr></table>
<TABLE BORDER=0 CELLSPACING=2 CELLPADDING=2 WIDTH=650>
<tr>
<th colspan=10 align=center bgcolor="#666699"><font size=4
color=""white"">Override Payment Days</th></tr>
<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left width=650>&nbsp;&nbsp;&nbsp;&nbsp;Please
enter a Customer Code and the Payment Days override data:</th></tr>
<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<td colspan=10 align=left width=650>&nbsp;&nbsp;&nbsp;&nbsp;Customer
Code &nbsp;&nbsp;**CustCode2**&nbsp;&nbsp;Override Payment Days&nbsp;
&nbsp;**PaymentDays**
&nbsp;&nbsp;**Add**&nbsp;&nbsp;**Update**&nbsp;&nbsp;**Delete**</td>
</tr>

```

```

<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<td colspan=10 align=center width=650><font color=""red"">
**ErrorMessage**</font></td></tr>
<tr>
<td colspan=10 width=650>&nbsp;</td></tr>
<tr>
<td colspan=5 align=left>**CreditHomePg**</td>
<td colspan=5 align=right>**WOSMainSel**</td></tr>

</TABLE>

**SPIDERSESSION**
[[/SPIDERPAGE]]

</BODY>
</HTML>

```

xvii.

```

pgCreditInstruct.htm
<HTML>
<HEAD>
<TITLE>Credit Instructions</TITLE>
</HEAD>
[[SPIDERPAGE NAME=pgCreditInstruct]]

<center>
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=10 align=center>**logobeth**</td></tr>

<tr><td colspan=10 align=center>&nbsp;</td></tr>
</TABLE>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=800>
<tr>
<td colspan=10 align=center bgcolor="#666699">&nbsp;</td></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 ><font size=5><i>As an existing customer, if you
desire to use Bethlehem Steel as<br>a credit reference, you will
need to participate in our electronic credit system:</i></font></th>
</tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<tr><td colspan=10 align=center bgcolor="#666699">&nbsp;</td></tr>
</table>

<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr><td colspan=10 align=left>&nbsp;</td></tr>
<tr><td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left><font size=4>(1) - Access 'Customer
Corner' (from Bethlehem's Home Page at
<br> www.bethsteel.com - 'Customers' - or use the link below.)</font>
</th></tr>
<tr>
```

```

<td colspan=10 align=left>&nbsp;</td></tr>
<tr>
<th colspan=10 align=left><font size=4>(2) - Request Password (if
needed) by clicking on 'Need Password?'.
<br> After receiving your logon id and password via email,
return to the<br> Customer Corner as stated in (1) above.</font>
</th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<th colspan=10 align=left><font size=4>(3) - Click 'Customer Log In'.
</font>
</th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<th colspan=10 align=left><font size=4>(4) - Once you proceed beyond
the ""login"" screen, choose the link titled<br>
""Use Bethlehem as a Trade Reference"". </font>
</th></tr>
<tr>
<td colspan=10 align=left>&nbsp;</td></tr>
<th colspan=10 align=left><font size=4>(5) - Follow the instructions
for choosing and setting a ""CIN"" number.</font>
</th></tr>

</TABLE>

<br>
<br>
<table BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH=650>
<tr>
<td colspan=2 align=left><font color=blue>**GoToCustCorner**</font>
</td>
<td colspan=6 >&nbsp;&nbsp;</td>
<td colspan=2 align=right><font color=blue>**BackToCreditHomePage**</font>
</td>
</tr>
</TABLE>

<BR>
**SPIDERSESSION**
[[/SPIDERPAGE]]
```

</BODY>
</HTML>

xviii.

```

pgError.htm
<HTML>
<HEAD>
<TITLE>Error Handling</TITLE>
</HEAD>
<BODY>
<CENTER>
[[SPIDERPAGE NAME=pgError]]
```



```

<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
**SqlServerError**</font></td>
</tr><tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
**LogonAgain**</font></td>
</tr><tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
</font></td>
</tr><tr>
<td align=RIGHT valign=middle><font size=3><B></B></td>
<td align=LEFT valign=middle><font size=2><font color=""#FF0000"">
</font></td>
</tr>
</table>
<BR>
**SPIDERSESSION**
[[/SPIDERPAGE]]
</CENTER>
</font>
</BODY>
</HTML>

```

XII. JAVA Source Files

i.

```

AppProcessPg.java
// This file has been generated by NetDynamics Studio on Mon Jun. 19,
2000 16:03

```

```

package CreditApp;
```

```

import java.awt.event.*;
import spider.event.*;
import spider.database.*;
import spider.visual.*;
```

```

//THIS PAGE SHOWS UP IF THE USER'S APPLICATION WAS SUCESSFULLY SENT
TO THE EMAIL ACCOUNT

```

```

//[[SPIDER_CLASS BEGIN
public class AppProcessPg extends spider.visual.CSpPage
//]]SPIDER_CLASS END
{
    //[[SPIDER_EVENTS BEGIN
        //[[SPIDER_EVENT<GoToCreditHmPg_onWebEvent>
        public int GoToCreditHmPg_onWebEvent(CSpWebEvent
event)
        {
            return load("CreditDeptHomePg");
        }
    //]]SPIDER_EVENT<GoToCreditHmPg_onWebEvent>
}
//]]SPIDER_EVENTS END
}
```

ii.

```

AppResults.java
// This file has been generated by NetDynamics Studio on Thu Jun. 15,
2000 08:50

```

```

package CreditApp;
```

```

import java.awt.event.*;
import spider.event.*;
```

```

import spider.database.*;
import spider.visual.*;
import spider.util.*;
import spider.*;
import spider.session.*;
import java.lang.*;

//[[SPIDER_CLASS BEGIN
public class AppResults extends spider.visual.CSpPage
//]]SPIDER_CLASS END
{
    private transient Credit the_CreditObject;
    //[[SPIDER_EVENTS BEGIN

        //[[SPIDER_EVENT<this_onBeforeLoadEvent>
        public int this_onBeforeLoadEvent(CSpPageEvent event)
        {
            the_CreditObject = (Credit)
Credit.getInstance();
            return (PROCEED);
        }
    //]]SPIDER_EVENT<this_onBeforeLoadEvent>

        //[[SPIDER_EVENT
<IncorporateName_onBeforeDisplayEvent>
        public int IncorporateName_onBeforeDisplayEvent
(CSpDisplayEvent event)
        {
            //The code below(all of it) checks to see if the
field is null, if it is then the
            //field will be displayed on the AppResults page for
the user to fill in again.

            int x =
the_CreditObject.getIncorporatedName().trim().length();
            if( x != 0)
                return SKIP;
            else
                return PROCEED;
        }
    //]]SPIDER_EVENT
<IncorporateName_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<CpnyNameLabel_onBeforeDisplayEvent>
        public int CpnyNameLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
        {
            int x =
the_CreditObject.getIncorporatedName().trim().length();
            if( x != 0)
                return SKIP;
            else
                return PROCEED;
        }
    //]]SPIDER_EVENT<CpnyNameLabel_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<IncorpName_onBeforeDisplayEvent>
        public int IncorpName_onBeforeDisplayEvent
(CSpDisplayEvent event)
        {
            int x =

```

```

the_CreditObject.getIncorporatedName().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<IncorpName_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadAddLabel_onBeforeDisplayEvent>
    public int HeadAddLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getHeadquarterAddress().trim().length();
    if( a != 0)
    {
        int b =
the_CreditObject.getHeadquarterCity().trim().length();
        if( b != 0)
        {
            int c =
the_CreditObject.getHeadquarterState().trim().length();
            if( c != 0)
            {
                int d =
the_CreditObject.getHeadquarterZip().trim().length();
                if( d != 0)
                {
                    int e =
the_CreditObject.getHeadquarterPhone().trim().length();
                    if( e != 0)
                    {
                        return SKIP;
                    }
                    else
                    {
                        return PROCEED;
                    }
                }
                else
                {
                    return PROCEED;
                }
            }
            else
            {
                return PROCEED;
            }
        }
        else
        {
            return PROCEED;
        }
    }
    else
    {
        return PROCEED;
    }
}
//]]SPIDER_EVENT<HeadAddLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadStAdd_onBeforeDisplayEvent>
    public int HeadStAdd_onBeforeDisplayEvent
(CSpDisplayEvent event)

```

```

        {
            int x =
the_CreditObject.getHeadquarterAddress().trim().length();
            if( x != 0)
                return SKIP;
            else
                return PROCEED;
        }
    //]]SPIDER_EVENT<HeadStAdd_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<HeadStLabel_onBeforeDisplayEvent>
    public int HeadStLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterAddress().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<HeadStLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<HeadquCity_onBeforeDisplayEvent>
    public int HeadquCity_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterCity().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<HeadquCity_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<HeadCityLabel_onBeforeDisplayEvent>
    public int HeadCityLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterCity().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<HeadCityLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<HeadState_onBeforeDisplayEvent>
    public int HeadState_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterState().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<HeadState_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<HeadquStLabel_onBeforeDisplayEvent>
    public int HeadquStLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)

```

```

    {
        int x =
the_CreditObject.getHeadquarterState().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
    }
//]]SPIDER_EVENT<HeadquStLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadZip_onBeforeDisplayEvent>
public int HeadZip_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterZip().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadZip_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquZipLabel_onBeforeDisplayEvent>
public int HeadquZipLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterZip().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadquZipLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquPh_onBeforeDisplayEvent>
public int HeadquPh_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterPhone().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadquPh_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadPhLabel_onBeforeDisplayEvent>
public int HeadPhLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getHeadquarterPhone().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<HeadPhLabel_onBeforeDisplayEvent>
```

```

        //[[SPIDER_EVENT<DunFedLabel_onBeforeDisplayEvent>
        public int DunFedLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
the_CreditObject.getDun2().trim
().length() !=0 &&
the_CreditObject.getDun3().trim
().length() !=0)
{
    if(the_CreditObject.getFedID1
().trim().length() !=0 &&
the_CreditObject.getFedID2
().trim().length() !=0 )
        return SKIP;
    else
        return (PROCEED);
}
else
    return (PROCEED);
}
//]]SPIDER_EVENT<DunFedLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Duns1_onBeforeDisplayEvent>
public int Duns1_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
the_CreditObject.getDun2().trim
().length() !=0 &&
the_CreditObject.getDun3().trim
().length() !=0)
        return SKIP;
    else
        return (PROCEED);
}
//]]SPIDER_EVENT<Duns1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Duns2_onBeforeDisplayEvent>
public int Duns2_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
the_CreditObject.getDun2().trim
().length() !=0 &&
the_CreditObject.getDun3().trim
().length() !=0)
        return SKIP;
    else
        return (PROCEED);
}
//]]SPIDER_EVENT<Duns2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Duns3_onBeforeDisplayEvent>
public int Duns3_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    if(the_CreditObject.getDun1().trim
().length() !=0 &&
the_CreditObject.getDun2().trim
()
)
}

```

```

().length() !=0 &&
                           the_CreditObject.getDun3().trim
().length() !=0)
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<Duns3_onBeforeDisplayEvent>

//[[SPIDER_EVENT<DunsLabel_onBeforeDisplayEvent>
public int DunsLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
if(the_CreditObject.getDun1().trim
().length() !=0 &&
                           the_CreditObject.getDun2().trim
().length() !=0 &&
                           the_CreditObject.getDun3().trim
().length() !=0)
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<DunsLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Fed1_onBeforeDisplayEvent>
public int Fed1_onBeforeDisplayEvent (CSpDisplayEvent
event)
{
if(the_CreditObject.getFedID1().trim
().length() !=0 &&
                           the_CreditObject.getFedID2().trim
().length() !=0 )
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<Fed1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Fed2_onBeforeDisplayEvent>
public int Fed2_onBeforeDisplayEvent (CSpDisplayEvent
event)
{
if(the_CreditObject.getFedID1().trim
().length() !=0 &&
                           the_CreditObject.getFedID2().trim
().length() !=0 )
                           return SKIP;
else
                           return (PROCEED);
}
//]]SPIDER_EVENT<Fed2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<FedLabel_onBeforeDisplayEvent>
public int FedLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
if(the_CreditObject.getFedID1().trim
().length() !=0 &&
                           the_CreditObject.getFedID2().trim
().length() !=0 )
                           return SKIP;
else
                           return (PROCEED);
}

```

```

        }
    //]]SPIDER_EVENT<FedLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankRefLabel_onBeforeDisplayEvent>
    public int BankRefLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a = the_CreditObject.getBankName
().trim().length();
    if( a != 0)
    {
        int b =
the_CreditObject.getBankStreet().trim().length();
        if( b != 0)
        {
            int c =
the_CreditObject.getBankCity().trim().length();
            if( c != 0)
            {
                int d =
the_CreditObject.getBankState().trim().length();
                if( d != 0)
                {
                    int e =
the_CreditObject.getBankZip().trim().length();
                    if( e != 0)
                        return SKIP;
                    else
                        return PROCEED;
                }
                else
                    return PROCEED;
            }
            else
                return PROCEED;
        }
        else
            return PROCEED;
    }
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankRefLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankName1_onBeforeDisplayEvent>
    public int BankName1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankName
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankName1_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankNameLb_onBeforeDisplayEvent>
    public int BankNameLb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankName
().trim().length();
    if( x != 0)

```

```

                return SKIP;
            else
                return PROCEED;
        }
    //]]SPIDER_EVENT<BankNameLb_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankStLabel_onBeforeDisplayEvent>
    public int BankStLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getBankStreet().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankStLabel_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankStr_onBeforeDisplayEvent>
    public int BankStr_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getBankStreet().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankStr_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankCity1_onBeforeDisplayEvent>
    public int BankCity1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankCity
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankCity1_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BankCityLb_onBeforeDisplayEvent>
    public int BankCityLb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankCity
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
    //]]SPIDER_EVENT<BankCityLb_onBeforeDisplayEvent>
```

```

// [[SPIDER_EVENT<BankSt1_onBeforeDisplayEvent>
public int BankSt1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankState
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankSt1_onBeforeDisplayEvent>

// [[SPIDER_EVENT<BankStLb_onBeforeDisplayEvent>
public int BankStLb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankState
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankStLb_onBeforeDisplayEvent>

// [[SPIDER_EVENT<BankZip1_onBeforeDisplayEvent>
public int BankZip1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankZip
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankZip1_onBeforeDisplayEvent>

// [[SPIDER_EVENT<BankZipLb_onBeforeDisplayEvent>
public int BankZipLb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getBankZip
().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<BankZipLb_onBeforeDisplayEvent>

// [[SPIDER_EVENT<TradeRefLabel_onBeforeDisplayEvent>
public int TradeRefLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName1().trim().length();
    if( a != 0)
    {
        int b =
the_CreditObject.getTradeRefCityStZip1().trim().length();

```

```

        if( b != 0 )
        {
            int c =
the_CreditObject.getTradeRefName2().trim().length();
            if( c != 0 )
            {
                int d =
the_CreditObject.getTradeRefCityStZip2().trim().length();
                if( d != 0 )
                {
                    int e =
the_CreditObject.getTradeRefName3().trim().length();
                    if( e != 0 )
                    {
                        int f =
the_CreditObject.getTradeRefCityStZip3().trim().length();
                        if( f != 0 )
                            return SKIP;
                        else
                            return
PROCEED;
                    }
                    else
                        return PROCEED;
                }
                else
                    return PROCEED;
            }
            else
                return PROCEED;
        }
        else
            return PROCEED;
    }
//]]SPIDER_EVENT<TradeRefLabel_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref1Lab_onBeforeDisplayEvent>
public int Ref1Lab_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName1().trim().length();
    if( a != 0 )
    {
        int b =
the_CreditObject.getTradeRefCityStZip1().trim().length();
        if( b != 0 )
            return SKIP;
        else
            return PROCEED;
    }
    else
        return PROCEED;
}
//]]SPIDER_EVENT<Ref1Lab_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref1NameLabel_onBeforeDisplayEvent>
public int Ref1NameLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =

```

```

the_CreditObject.getTradeRefName1().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefName1_onBeforeDisplayEvent>
public int RefName1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x = the_CreditObject.getTradeRefName1().trim
().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref1CityLabel_onBeforeDisplayEvent>
public int Ref1CityLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip1().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefCity1_onBeforeDisplayEvent>
public int RefCity1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip1().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref2Lab_onBeforeDisplayEvent>
public int Ref2Lab_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName2().trim().length();
    if( a != 0)
    {
        int b =
the_CreditObject.getTradeRefCityStZip2().trim().length();
        if( b != 0)
            return SKIP;
        else
            return PROCEED;
    }
    else

```

```

                    return PROCEED;;
                }
            //]]SPIDER_EVENT<Ref2Lab_onBeforeDisplayEvent>

            //[[SPIDER_EVENT<Ref2NameLabel_onBeforeDisplayEvent>
            public int Ref2NameLabel_onBeforeDisplayEvent
            (CSpDisplayEvent event)
            {
                int x =
the_CreditObject.getTradeRefName2().trim().length();
                if( x != 0)
                    return SKIP;
                else
                    return PROCEED;
            }
        //]]SPIDER_EVENT<RefName2_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<RefName2_onBeforeDisplayEvent>
        public int RefName2_onBeforeDisplayEvent
        (CSpDisplayEvent event)
        {
            int x =
the_CreditObject.getTradeRefName2().trim().length();
            if( x != 0)
                return SKIP;
            else
                return PROCEED;
        }
    //]]SPIDER_EVENT<RefName2_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<Ref2CityLabel_onBeforeDisplayEvent>
    public int Ref2CityLabel_onBeforeDisplayEvent
    (CSpDisplayEvent event)
    {
        int x =
the_CreditObject.getTradeRefCityStZip2().trim().length();
        if( x != 0)
            return SKIP;
        else
            return PROCEED;
    }
}]]SPIDER_EVENT<RefCity2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<RefCity2_onBeforeDisplayEvent>
public int RefCity2_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip2().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
}]]SPIDER_EVENT<RefCity2_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Ref3Lab_onBeforeDisplayEvent>
public int Ref3Lab_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int a =
the_CreditObject.getTradeRefName3().trim().length();
    if( a != 0)

```

```

        {
            int b =
the_CreditObject.getTradeRefCityStZip3().trim().length();
            if( b != 0)
                return SKIP;
            else
                return PROCEED;
        }
    else
        return PROCEED;
}
//]]SPIDER_EVENT<Ref3Lab_onBeforeDisplayEvent>

// [[SPIDER_EVENT<Ref3NameLabel_onBeforeDisplayEvent>
public int Ref3NameLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefName3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName3_onBeforeDisplayEvent>

// [[SPIDER_EVENT<RefName3_onBeforeDisplayEvent>
public int RefName3_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefName3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefName3_onBeforeDisplayEvent>

// [[SPIDER_EVENT<Ref3CityLabel_onBeforeDisplayEvent>
public int Ref3CityLabel_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity3_onBeforeDisplayEvent>

// [[SPIDER_EVENT<RefCity3_onBeforeDisplayEvent>
public int RefCity3_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    int x =
the_CreditObject.getTradeRefCityStZip3().trim().length();
    if( x != 0)
        return SKIP;
    else
        return PROCEED;
}
//]]SPIDER_EVENT<RefCity3_onBeforeDisplayEvent>
```

```

// [[SPIDER_EVENT<Submit_onWebEvent>
public int Submit_onWebEvent(CSpWebEvent event)
{
    Credit the_CreditObject =
Credit.getInstance();

        /* setting all the fields on this page so when
the user hits the "submit" button
            their information is saved */
        the_CreditObject.setIncorporateName
(getDisplayField("IncorporateName").getValue().toString());
        the_CreditObject.setHeadStAdd
(getDisplayField("HeadStAdd").getValue().toString());
        the_CreditObject.setHeadquCity
(getDisplayField("HeadquCity").getValue().toString());
        the_CreditObject.setHeadState(getDisplayField
("HeadState").getValue().toString());
        the_CreditObject.setHeadZip
(getDisplayField("HeadZip").getValue().toString());
        the_CreditObject.setHeadquPh
(getDisplayField("HeadquPh").getValue().toString());
        //        the_CreditObject.setHeadquFax
(getDisplayField("HeadquFax").getValue().toString());
        //        the_CreditObject.setHeadquWww
(getDisplayField("HeadquWww").getValue().toString());
        the_CreditObject.setDuns1
(getDisplayField("Duns1").getValue().toString());
        the_CreditObject.setDuns2
(getDisplayField("Duns2").getValue().toString());
        the_CreditObject.setDuns3
(getDisplayField("Duns3").getValue().toString());
        the_CreditObject.setFed1
(getDisplayField("Fed1").getValue().toString());
        the_CreditObject.setFed2
(getDisplayField("Fed2").getValue().toString());
        the_CreditObject.setBankName1
(getDisplayField("BankName1").getValue().toString());
        the_CreditObject.setBankStr
(getDisplayField("BankStr").getValue().toString());
        the_CreditObject.setBankCity1
(getDisplayField("BankCity1").getValue().toString());
        the_CreditObject.setBankSt1
(getDisplayField("BankSt1").getValue().toString());
        the_CreditObject.setBankZip1
(getDisplayField("BankZip1").getValue().toString());
        //        the_CreditObject.setAcctNm
(getDisplayField("AcctNm").getValue().toString());
        //        the_CreditObject.setAcctPhone
(getDisplayField("AcctPhone").getValue().toString());
        the_CreditObject.setRefName1
(getDisplayField("RefName1").getValue().toString());
        //        the_CreditObject.setRefStreet1
(getDisplayField("RefStreet1").getValue().toString());
        the_CreditObject.setRefCity1
(getDisplayField("RefCity1").getValue().toString());
        //        the_CreditObject.setAttnName1
(getDisplayField("AttnName1").getValue().toString());
        //        the_CreditObject.setRefPh1
(getDisplayField("RefPh1").getValue().toString());
        //        the_CreditObject.setRefFaxNum1
(getDisplayField("RefFaxNum1").getValue().toString());

```

```

        the_CreditObject.setRefName2
(getDisplayField("RefName2").getValue().toString());
        //          the_CreditObject.setRefStreet2
(getDisplayField("RefStreet2").getValue().toString());
        //          the_CreditObject.setRefCity2
(getDisplayField("RefCity2").getValue().toString());
        //          the_CreditObject.setAttnName2
(getDisplayField("AttnName2").getValue().toString());
        //          the_CreditObject.setRefPh2
(getDisplayField("RefPh2").getValue().toString());
        //          the_CreditObject.setRefFaxNum2
(getDisplayField("RefFaxNum2").getValue().toString());
        //          the_CreditObject.setRefName3
(getDisplayField("RefName3").getValue().toString());
        //          the_CreditObject.setRefStreet3
(getDisplayField("RefStreet3").getValue().toString());
        //          the_CreditObject.setRefCity3
(getDisplayField("RefCity3").getValue().toString());
        //          the_CreditObject.setAttnName3
(getDisplayField("AttnName3").getValue().toString());
        //          the_CreditObject.setRefPh3
(getDisplayField("RefPh3").getValue().toString());
        //          the_CreditObject.setRefFaxNum3
(getDisplayField("RefFaxNum3").getValue().toString());

        //
        //          if(! the_CreditObject.isValid())
        //          return load("AppResults");

        //BethSteel set up this email account
String to =
""bscredit@bethsteel.com";
//      String to = "vicky.alban@eds.com";
String from = getDisplayFieldValue
("From").toString();
if(from.indexOf("@",0) == -1 ||
from.indexOf(".",0) == -1)
{
    from =
"mailServer@bsco.com";
}
String host = "mail.bsco.com";
String host = "10.50.1.1";
String subject = "Process Credit
Application";

        //this is the code that will be
displayed on the emial sent to the above address
        String msgText = "" Past Customer:
"" + the_CreditObject.getPastCust()           + "\n What Year: "
+ the_CreditObject.getPastYear()             + "\n Name Used: "
+ the_CreditObject.getPastName()            + "\n Product: "
+ the_CreditObject.getPlate() + " " +
the_CreditObject.getSemiFinished() + " " +
the_CreditObject.getPipe() + " " + the_CreditObject.getSheet() +
" " + the_CreditObject.getRail() + " " +
the_CreditObject.getReclamation() + " " +
the_CreditObject.getTinPlate() + " " + the_CreditObject.getBar() +
" " +
the_CreditObject.getSecondary() + " " +
the_CreditObject.getOther() + " " +
the_CreditObject.getOtherSpec()

```

```

+ ""\n Business
Division: "" + the_CreditObject.getBusinessDivision()
+ ""\n Domestic: "" +
the_CreditObject.getDomestic()
+ ""\n"
+ ""\n Company
Information: ""
+ ""\n Company Name:
"" + the_CreditObject.getIncorporatedName() +
the_CreditObject.getIncorporateName()
+ ""\n Company
Division Name: "" + the_CreditObject.getCpnyDivisionName() + ""\n PO
Name: "" + the_CreditObject.getPOName()
+ ""\n"
+ ""\n Headquarter
Address: ""
+ ""\n Company
Headquarter Street Address: "" +
the_CreditObject.getHeadquarterAddress() +
the_CreditObject.getHeadStAdd()
+ ""\n Company
Headquarter Box Number: "" + the_CreditObject.getHeadquarterBoxNum()
+ ""\n Company
Headquarter City: "" + the_CreditObject.getHeadquarterCity() +
the_CreditObject.getHeadquCity()
+ ""\n Company
Headquarter Zip: "" + the_CreditObject.getHeadquarterZip() +
the_CreditObject.getHeadZip()
+ ""\n Company
Headquarter State: "" + the_CreditObject.getHeadquarterState() +
the_CreditObject.getHeadState()
+ ""\n Company
Headquarter Foreign Address (if applicable): "" +
the_CreditObject.getHeadquarterForeignAddress()
+ ""\n Company
Headquarter Phone Number: "" + the_CreditObject.getHeadquarterPhone()
+ the_CreditObject.getHeadquPh()
+ ""\n Company
Headquarter Fax Number: "" + the_CreditObject.getHeadquarterFax() +
the_CreditObject.getHeadquFax()
+ ""\n Company
Headquarter Web Address: "" + the_CreditObject.getHeadquarterWeb() +
the_CreditObject.getHeadquWWw()
+ ""\n"
+ ""\n Branch
Location Information: ""
+ ""\n Will they
Purchase through a branch location?: "" +
the_CreditObject.getPurchaseThroughBranch()
+ ""\n Branch Name:
"" + the_CreditObject.getBranchName() + ""\n Branch Division: "" +
the_CreditObject.getBranchDivision()
+ ""\n Branch Street
Address: "" + the_CreditObject.getBranchStAddress()
+ ""\n Branch City:
"" + the_CreditObject.getBranchCity() + ""\n Branch State: "" +
the_CreditObject.getBranchState()
+ ""\n Branch Zip:
"" + the_CreditObject.getBranchZip() + ""\n Branch Foreign Address:
"" + the_CreditObject.getBranchForeignAddress()
+ ""\n Branch Phone
Number: "" + the_CreditObject.getBranchPhone() + ""\n Branch Fax
Number: "" + the_CreditObject.getBranchFax()
//+ ""\n Branch
Street Address: "" + the_CreditObject.getBranchStAddress()
+ ""\n"

```

```

+ ""\n Invoices and
Other Correspondence Information: ""
+ ""\n Invoices
should be sent to: "" + the_CreditObject.getInvoiceCorrespondence()
+ ""\n Other
locations Street Address: "" + the_CreditObject.getOtherStreet()
+ ""\n Other
locations Box Number: "" + the_CreditObject.getOtherBoxNum()
+ ""\n Other
locations City: "" + the_CreditObject.getOtherCity() + ""\n Other
locations state: "" + the_CreditObject.getOtherState()
+ ""\n Other
locations Zip Code: "" + the_CreditObject.getOtherZip() + ""\n
Foreign Address if applicable: "" +
the_CreditObject.getInvoiceForeignAddress()
+ ""\n"
+ ""\n Bethlehem
Sales Person Information: ""
+ ""\n Have you ever
contacted Bethlehem in the past?: "" +
the_CreditObject.getContactWithBeth()
+ ""\n If yes, please
provide the Contacts Name: "" + the_CreditObject.getContactBethName()
+ ""\n"
+ ""\n Accounts
Payable Information: ""
+ ""\n Accounts
Payable system is: "" + the_CreditObject.getApSystem() + ""\n
Payments are mailed to: "" + the_CreditObject.getApMailedTo()
+ ""\n Buyer's Name:
"" + the_CreditObject.getBuyersName() + ""\n Buyer's Phone Number:
"" + the_CreditObject.getBuyersPhoneNum()
+ ""\n Buyer's Email
Address: "" + the_CreditObject.getBuyersEmail() + ""\n AP Supervisors
Name: "" + the_CreditObject.getApSupervisorName()
+ ""\n AP Supervisors
Phone Number: "" + the_CreditObject.getApPhoneNum() + ""\n AP
Supervisors Email Address: "" + the_CreditObject.getApEmail()
+ ""\n Person to
contact for a financial statement: "" +
the_CreditObject.getFinancialSupportName()
+ ""\n"
+ ""\n DUNS # and Fed
I.D. Information: ""
+ ""\n DUNS Number:
"" + the_CreditObject.getDun1() + ""-"" + the_CreditObject.getDun2()
+ ""-"" + the_CreditObject.getDun3() + the_CreditObject.getDuns1() +
""-"" + the_CreditObject.getDuns2() + ""-"" +
the_CreditObject.getDuns3()
+ ""\n Fed ID: "" +
the_CreditObject.getFedID1() + ""-"" + the_CreditObject.getFedID2() +
the_CreditObject.getFed1() + ""-"" + the_CreditObject.getFed2()
+ ""\n"
+ ""\n Primary Bank
Reference Information: ""
+ ""\n Bank Name: ""
+ the_CreditObject.getBankName() + the_CreditObject.getBankName1() +
""\n Bank Street Address: "" + the_CreditObject.getBankStreet() +
the_CreditObject.getBankStr()
+ ""\n Bank City: ""
+ the_CreditObject.getBankCity() + the_CreditObject.getBankCity1() +
""\n Bank State: "" + the_CreditObject.getBankState() +
the_CreditObject.getBankSt1()
+ ""\n Banks Country:
"" + the_CreditObject.getBankOther() + ""\n Bank Zip Code: "" +
the_CreditObject.getBankZip() + the_CreditObject.getBankZip1()

```

```

        + ""\n Account
Officers Name: "" + the_CreditObject.getAcctOfficerName() +
the_CreditObject.getAcctNm() + ""\n Account Officers Phone Number:
"" + the_CreditObject.getAcctOfficerPhoneNum() +
the_CreditObject.getAcctPhone()
        + ""\n"
        + ""\n Trade

Reference Information: ""
        + ""\n Trade
Reference #1 Name: "" + the_CreditObject.getTradeRefName1() +
the_CreditObject.getRefName1() + ""\n Trade Reference #1 Street
Address: "" + the_CreditObject.getTradeRefStreet1() +
the_CreditObject.getRefStreet1()
        + ""\n Trade

Reference #1 City State and Zip: "" +
the_CreditObject.getTradeRefCityStZip1() +
the_CreditObject.getRefCity1() + ""\n Trade Reference #1 Attn: "" +
the_CreditObject.getTradeRefAttn1() + the_CreditObject.getAttnName1()
        + ""\n Trade

Reference #1 Phone Number: "" + the_CreditObject.getTradeRefPhone1() +
+ the_CreditObject.getRefPh1() + ""\n Trade Reference #1 Fax Number:
"" + the_CreditObject.getTradeRefFax1() +
the_CreditObject.getRefFaxNum1()
        + ""\n Trade

Reference #2 Name: "" + the_CreditObject.getTradeRefName2() +
the_CreditObject.getRefName2() + ""\n Trade Reference #2 Street: "" +
the_CreditObject.getTradeRefStreet2() +
the_CreditObject.getRefStreet2()
        + ""\n Trade

Reference #2 City State and Zip: "" +
the_CreditObject.getTradeRefCityStZip2() +
the_CreditObject.getRefCity2() + ""\n Trade Reference #2 Attn: "" +
the_CreditObject.getTradeRefAttn2() + the_CreditObject.getAttnName2()
        + ""\n Trade

Reference #2 Phone Number: "" + the_CreditObject.getTradeRefPhone2() +
+ the_CreditObject.getRefPh2() + ""\n Trade Reference #2 Fax Number:
"" + the_CreditObject.getTradeRefFax2() +
the_CreditObject.getRefFaxNum2()
        + ""\n Trade

Reference #3 Name: "" + the_CreditObject.getTradeRefName3() +
the_CreditObject.getRefName3() + ""\n Trade Reference #3 Street: "" +
the_CreditObject.getTradeRefStreet3() +
the_CreditObject.getRefStreet3()
        + ""\n Trade

Reference #3 City State and Zip: "" +
the_CreditObject.getTradeRefCityStZip3() +
the_CreditObject.getRefCity3() + ""\n Trade Reference #3 Attn: "" +
the_CreditObject.getTradeRefAttn3() + the_CreditObject.getAttnName3()
        + ""\n Trade

Reference #3 Phone Number: "" + the_CreditObject.getTradeRefPhone3() +
+ the_CreditObject.getRefPh3() + ""\n Trade Reference #3 Fax Number:
"" + the_CreditObject.getTradeRefFax3() +
the_CreditObject.getRefFaxNum3();

```

```

//myCart =
Cart.getCartFromUserSession();
//Hashtable userCart = myCart.getCart
();
//String[] userItem;

/*for (Enumeration e =
userCart.elements(); e.hasMoreElements();) */

```

```

        {
            userItem = (String[]) e.nextElement
        );
        for(int i=0;i<userItem.length;i++)
        {
            if
            (userItem[i].equals(""\u00a0; ""))
            userItem[i] = """;;
        }
        msgText = msgText +
        ""\n          Key: "" + userItem[0]
        + ""\n
        Quarter: "" + userItem[1]
        + ""\n
        Year: "" + userItem[2]
        +
        ""\nAuthor/Inventor: "" + userItem[3] + ""\n      Title: "" +
        userItem[4]
        + ""\n      Patent
        No: "" + userItem[5] + ""\n\n";
    } */
    EmailObj the_Mailer = new EmailObj();
    the_Mailer.sendMail(host, from, to,
    subject, msgText);
}

```

```

    Credit.saveInstance
    (the_CreditObject);
    //return(command);
    return load("AppProcessPg");
}
//]]SPIDER_EVENT<Submit_onWebEvent>

```

//]]SPIDER_EVENTS END

}

iii.

CBethIMSDataField.java
package ImsUtilities;

```

import spider.database.*;
import spider.session.*;
import java.lang.*;
import spider.util.*;

public abstract class CBethIMSDataField extends CSpDataField
{
    public static final boolean LOCAL_DEBUG = false;

    public static final int NO_PADDING = 0;
    public static final int FILLER_ONLY = 1;
    public static final int PAD_LEFT = 2;
    public static final int PAD_RIGHT = 3;

    public static final int NO_CASE_CHANGE = 0;
    public static final int UPPER_CASE = 1;
    public static final int LOWER_CASE = 2;

    public static final char DEFAULT_FILLER_CHARACTER =

```

```

' ';

protected void invalidateField(String pReason)
{
    CBethIMSProcedure the_parent =
(CBethIMSProcedure)getParent();
    CSpLog.send
(this,CSpLog.WARNING,pReason);
    the_parent.invalidateParameters();
}

public int afterInit()
{
    int command = super.afterInit();
    if(PROCEED != command)
        return command;
    //
    // Override the Studio Properties
    //
    setOnEmptyValue
(ON_EMPTY_VALUE_EXCLUDE);
    return PROCEED;
}

public abstract int getPaddingType();
public abstract int getCaseChange();

public char getFillerCharacter()
{
    return DEFAULT_FILLER_CHARACTER;
}

//
// derived classes MAY override this method to
provide specialized behavior
// which CSpDataField and CBethIMSDataField do not
provide
//
protected CSpValue getIMSValue()
{
    if(LOCAL_DEBUG)
        CSpLog.send
(this,CSpLog.USER_DEBUG,""CBethIMSDataField.getIMSValue() entering
super.getValue()"");
    CSpValue the_value = super.getValue
();
    if(LOCAL_DEBUG)
        CSpLog.send
(this,CSpLog.USER_DEBUG,""CBethIMSDataField.getIMSValue() return from
super.getValue()"");
    return the_value;
}

//
// This method is final so that derived classes
enforce the mandatory
// formatting which CBethIMSDataField provides.
getValue() behavior MAY
// be directed by overriding the getFillerCharacter()
method and MUST be
// directed by providing concrete method
getPaddingType()
//
// The getValue() behavior may be skipped entirely by
having method
// getPaddingType() return

```

```

CBethIMSDataField.NO_PADDING
    //
    public final CSpValue getValue()
    {
        if(LOCAL_DEBUG)
            CSpLog.send
(this,CSpLog.USER_DEBUG,""BEGIN Data Object [" + getParent().getName()
+ ""] getValue()" + getName());
        if(!(((CBethIMSPprocedure)getParent
()).areParametersValid()))
        {
            //
            // if someone else
already invalidated then stop processing
            //
            return new CSpNull();
        }
        int the_padding_type = getPaddingType
();
        int the_case_change = getCaseChange
();
        //
        // Allow skipping default behavior
        //
        if( (NO_PADDING == the_padding_type)
&&
        (NO_CASE_CHANGE ==
the_case_change) )
        {
            if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""NO_PADDING and
NO_CASE_CHANGE");
            if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() "");
            return getIMSValue();
        }
        else if( (NO_PADDING ==
the_padding_type) &&
        (LOWER_CASE ==
the_case_change) )
        {
            if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""JUST LOWER CASE");
            CSpValue the_value =
getIMSValue();
            if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() ");
            return new CSpString
(the_value.stringValue().toLowerCase());
        }
        else if( (NO_PADDING ==
the_padding_type) &&
        (UPPER_CASE ==
the_case_change) )
        {
            if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""JUST UPPER CASE");
            CSpValue the_value =
getIMSValue();
            if(LOCAL_DEBUG)

```

```

CSpLog.send(this,CSpLog.USER_DEBUG, ""END getValue() "");
                                                 return new CSpString
(the_value.stringValue().toUpperCase());
}

// Allocate buffer of exact size
int the_column_length =
getColumnLength();
StringBuffer the_string_buffer = new
StringBuffer(the_column_length);
char the_filler_character =
getFillerCharacter();

if(LOCAL_DEBUG)
CSpLog.send
(this,CSpLog.USER_DEBUG, "ColumnLength " + the_column_length);

switch (the_padding_type)
{
    case FILLER_ONLY :
    {
        if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG, "'FILLER_ONLY'");
for
(int i=0;i<the_column_length;i++)
the_string_buffer.append(the_filler_character);

break;
    }
    case PAD_LEFT :
    case PAD_RIGHT :
    {

CSpValue the_value = getIMSValue();
if(!
(((CBethIMSPprocedure)getParent()).areParametersValid()))
{

// if getIMSValue() invalidated the field
//

return new CSpNull();
}
if
(null == the_value)
{
invalidateField("'"CBethIMSDataField does not allow getIMSValue() =
null, returning CSpNull to get past'");
}

return new CSpNull();
}
if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG, "'getIMSValue() = '" +
the_value.stringValue());

```

```

String the_string = the_value.stringValue().trim();
if
(the_string.length() > the_column_length)
{

CSpLog.send(this,CSpLog.ERROR,
""Trimmed string size of getIMSValue() is greater than Data Fields
Column Length"");

the_string_buffer.append(the_string.toCharArray
(),0,the_column_length);
}
else
{

int the_padding_length = the_column_length - the_string.length();

if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""PaddingLength "" +
the_padding_length);

if(0 == the_padding_length)

the_string_buffer.append(the_string);

else if(PAD_LEFT == the_padding_type)

{

if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""PAD_LEFT"");

for(int i=0;i<the_padding_length;i++)

the_string_buffer.append(the_filler_character);

the_string_buffer.append(the_string);

}

else // PAD_RIGHT

{

if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""PAD_RIGHT"");

the_string_buffer.append(the_string);

for(int i=0;i<the_padding_length;i++)

the_string_buffer.append(the_filler_character);

}

}

break;
}
default :
{
}
// same as NO_PADDING for unknown Padding Type

```

```

CSpLog.send(this,CSpLog.WARNING,
        ""Invalid Padding Type "" + the_padding_type);

return getIMSValue();
    }
}
if(LOWER_CASE == the_case_change)
{
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""LOWER CASE");
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() "");
    return new CSpString
(the_string_buffer.toString().toLowerCase());
}
else if(UPPER_CASE ==
the_case_change)
{
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""UPPER CASE");
    CSpValue the_value =
getIMSValue();
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() ");
    return new CSpString
(the_string_buffer.toString().toUpperCase());
}
else
{
    if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""END getValue() ");
    return new CSpString
(the_string_buffer.toString());
}
}

CBethIMSFiller.java
iv.

package ImsUtilities;

public class CBethIMSFiller extends CBethIMSDataField
{
    public int getCaseChange()
    {
        return NO_CASE_CHANGE;
    }

    public int getPaddingType()
    {
        return NO_PADDING;
    }
}

CBethIMSPadRight.java
v.

package ImsUtilities;

public class CBethIMSPadRight extends CBethIMSDataField
{

```

```

        public int getCaseChange()
        {
            return NO_CASE_CHANGE;
        }

        public int getPaddingType()
        {
            return PAD_RIGHT;
        }
    }

CBethIMSProcedure.java
package ImsUtilities;

import spider.database.*;
import spider.util.*;
import java.util.*;
import spider.event.*;
import spider.session.*;

/**
 * An abstract base class for all Precise IMS PAC Data Objects.
 * The NetDynamics PAC technology and specifically the Precise IMS
PAC uses a
 * stored procedure interface to IMS transactions on MVS. For the
first release
 * of the Precise IMS PAC, only simple <b>string</b> input and output
parameters
 * are allowed in the procedure Metadata. Refer to Precise Manual.
For example:
*
* <pre>
* struct outputCE4ZWB01 {
*     string(12)    outMillOrderNumber;
*     string(26)    outPurchaseOrderNumber;
*     string(8)     outProductID;
*     string(6)     outThickness;
*     string(8)     outWidth;
*     string(8)     outOrderedLBS;
*     string(8)     outAvailableLBS;
*     string(8)     outUnshippedLBS;
*     string(8)     outShippedLBS;
*     string(2)     outPlantCode;
* }
*
* procedure CE4ZWB01 {
*     text ""Queries Light Flat Roll Mill Orders"";
*     in string(7)   inShorty;
*     in string(35)  inPono;
*     in string(20)  inCID;
*     in string(11)  inProduct;
*     in string(01)  inType;
*     in string(5)   inThickFrom;
*     in string(5)   inThickTo;
*     in string(7)   inWidthFrom;
*     in string(7)   inWidthTo;
*     in string(8)   inDesireDateFrom;
*     in string(8)   inDesireDateTo;
*     out struct outputCE4ZWB01[] outTable;
* }
* </pre>
*
* The IMS transactions, in the case of COBOL programs, have a input
and output
* buffer interface. Single-segment IMS transactions have a single
output buffer.

```

vi.

```

* Multi-segment IMS transactions have one or more output buffers
(i.e. like
* multiple rows coming back from SQL or RDBMS stored procedure).
The input and
* output buffers are bounded character buffers. These buffers
provide the
* interface to the IMS Message Queue. The schema of the input and
output buffers
* are specified in the ""01"" level in the COBOL program. The
""01"" level of the
* COBOL program is key in the development of the Metadata for the
IMS PAC. The
* Metadata schema must match the ""01"" level in the COBOL exactly.
The simple
* <b>string</b> restriction on parameters is required to support a
high speed
* marshalling of data between the NetDynamics PAC service and IMS.
*
* In addition to the limitation that only simple <b>string</b>
parameter types
* may be used, there is no PRE or POST processing to the input or
output parameters
* except to convert to and from EBCDIC and Big/Little Endian (see
Precise Manual
* for details on the <code>environ</code> column in the <b>appltab
</b> file.
*
* Currently, there is no automatic padding or truncation of the
input parameters
* in the IMS PAC service before they are packed into an input buffer
for the
* IMS Message Queue. Therefore, the out-of-the-box Precise IMS PAC
requires the
* developer to make sure that the input parameters are formatted
properly.
*
* This class, and CBethIMSDataField class, work together to enhance
the features
* of the Precise IMS PAC to include pre-processing of input
parameters and error
* handling. All input data fields (parameters) managed by this
class <b>must</b>
* be derived CBethIMSDataField classes.
*
* This class also enforces as Design Pattern which support compile-
time and
* run-time safe string references. The Design Pattern requires the
developer
* to include a template in their implementation of the derived class
as follows:
*
* <pre>
*   public static final String projectName = ""WebOrderStatus"";
*   public static final String dataObjectName = ""imsCE4ZWB01"";
*
*   public String getProjectName() // implementation of abstract
method
*   {
*       return projectName;
*   }
*
*   public String getDataObjectName() // implementation of abstract
method
*   {
*       return dataObjectName;
*   }

```

```

*
*   public static CBethIMSProcedure getInstance()
*   {
*       return (CBethIMSProcedure) CSpider.getDataObject(projectName
+ "."+ "" + dataObjectName);
*   }
* </pre>
*
* The result is that the developer may reference the name of the
class in a
* high performance and compile-time safe manner with the following:
*
* <code>CSpider.getDataObject(imsCE4ZWB01.dataObjectName);</code>
*
* The project name information support multi-project environments.
Please note
* that only two (2) of the five (5) class members in the template
are enforced
* by the superclass through abstract methods. Java Core Reflection
could be used
* to introspect the class to enforce the rest of the template.
*
* The <code>getInstance()</code> method is helper to get a reference
quickly.
*
* @see CBethIMSDataField
* @version Release: 1.0 Date: 1998/10/20
* @author Matthew J. Stevens, <a
href="http://www.objectstrategy.com">ObjectStrategy Inc</a>
*/
public abstract class CBethIMSProcedure extends CSpProcedure
{
    /**
     * Flag used as a condition for debug logging within
the class.
     * Since this member is static final, any decent
compiler should optimize out
     * any static condition. For example:
     *
     * <pre>
     * if(LOCAL_DEBUG) CSpLog.send
(this,CSpLog.USER_DEBUG,""x = "" + x);
     * </pre>
     *
     * is never included in the binary class file if
LOCAL_DEBUG==false.
     */
    public static final boolean LOCAL_DEBUG = true;

    /**
     * Support compile time and run-time safe string
references. Must be
     * implemented by developer in the derived class.
See class description above
     * for the recommended template. Recommended for use
in multi-project applications.
     * @return The name of the project (package) for this
class.
     */
    public abstract String getProjectName();

    /**
     * Support compile time and run-time safe string
references. Must be
     * implemented by developer in the derived class.
See class description above

```

```

        * for the recommended template.
        * @return The name of the class.
        */
    public abstract String getDataObjectName();

    /**
     // Override Spider superclass so that extra Invalid
Parameter Flag is reset
    /**
     public void clearAllValues()
    {
        resetParametersValidFlag();
        super.clearAllValues();
    }

    /**
     // overrided method from CSpFileBased which is the
first method to get called
     // after the File Based object is loaded and init'd.
     //
     // The superclass behaviour is to start the AfterInit
EVENT.  Here we ensure
     // that the superclass behaviour is called first,
then we
    /**
    /**
     * Add behavior to enable certain events
automatically and enforce design pattern.
     * This method should not be called by the developer;
it is called by
     * NetDynamics automatically.  This method is called
after the call to init().
     * This method is implemented to dispatch to
superclass first be added
     * behavior.  Therefore, the AFTER_INIT event will be
dispatched by the
     * superclass before the added behavior occurs.  This
method will force this
     * instance of the class to listen for the
DATA_OBJECT_EVENT's which are
     * automatically handled by the
processDataObjectEvent() method of this class.
     * Please note that this method is called
automatically by NetDynamics is three
     * cases: 1) when the object is loaded from disk
through the intrp model in
     * the CP Worker; 2) when the project graph is cloned
in the CP worker;
     * 3) when the Java client deserializes object in the
browser.
     * @return PROCEED on success or STOP when super
afterInit() fails or design pattern fails.
    */
    public int afterInit()
    {
        //
        // ensure Superclass behaviour called
first
        //
        int command = super.afterInit();
        if(PROCEED != command)
            return command;

        //
        // force the listening of the
DATA_OBJECT_EVENT so that overridden method
        // processDataObjectEvent() is called

```

```

always.

        // enableEvents needs to be included
every time there is an afterInit()
        //
        enableEvents
(CSpEvent.DATA_OBJECT_EVENT_MASK);

        //
        // Enforce design pattern that every
CBethIMSProcedure must use derived
        // CBethIMSDataFields for all INPUT
parameters. This forces the developer
        // to specify such in the Properties
of each INPUT parameter.
        //
        // Only need to do this at the server
(helps when using Java Client)
        //
        if(!isReadFromINTRP())
{
        if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""Not in first afterInit() so we
done now"");

        return PROCEED;
}

Enumeration the_fields =
getDataFields().elements();
while(the_fields.hasMoreElements())
{
        Object the_object;
        if(!((the_object =
the_fields.nextElement()) instanceof
CBethIMSDataField))
{
        if
(CSpDataField.PROCEDURE_IN_PARAM ==
((CSpDataField)the_object).getTableNameType())
{
        CSpLog.send(this,CSpLog.ERROR,
""IMS PAC Stored Procedure Input Data Field does not derive from
CBethIMSDataField [" +
the_object + ""] "");

        return STOP;
}
}
return PROCEED;
}

/**
 * Used internally to automatically handle Data
Object execution events for Comm/Booster condition determination.
*
* We are going to capture all CSpDataObjectEvent's
for this Data Object and
    * provide default behaviour for all IMS PAC Data
Objects. It appears that

```

```

        * this method gets called AFTER any derived class
        * <pre>
        * this_onDatabaseErrorEvent()
        * this_onBeforeExecuteEvent()
        * this_onAfterExecuteEvent()
        * </pre>
        * which is built in the Studio. For instance, if
the developer adds these
        * event handlers through the Studio Inspector, they
will be called BEFORE the
        * overridden method below. Therefore, if a
developer wants to customize the
        * the Database Events for a derived
CBethIMSPprocedure, they SHOULD override the
        * method below and dispatch to 'super'. The
developer may use
        * <pre>
        * this_onDatabaseErrorEvent()
        * this_onBeforeExecuteEvent()
        * this_onAfterExecuteEvent()
        * </pre>
        * but they will not get the behavior below first.
        *
        * Please Note: the CSppage Events for bound Data
Objects are either called
        * before or after the Data Object's Events. For
instance, the
        * <pre>
        * this_onBeforeDataObjectExecuteEvent()
        * </pre>
        * event is called BEFORE the Data Object's events.

However, the
        * <pre>
        * this_onAfterAllDataObjectsExecuteEvent()
        * this_onAfterDataObjectExecuteEvent()
        * this_onDataObjectErrorEvent()
        * </pre>
        * events are all called AFTER the Data Object's
events. Therefore, for
        * application error handling, the developer should
leverage the built-in
        * behaviour of the CBethIMSPprocedure by tapping into
the PAGE's
        * <pre>
        * this_onDataObjectErrorEvent()
        * </pre>
        * which is called AFTER the
CBethIMSPprocedure.processDataObjectEvent() method.
        * In this way, the 'source' Data Object of the Error
Event can be cast to
        * CBethIMSPprocedure and the getCommBoosterCondition
() method can be used to
        * decide what to display to the end user. Example,
if the Comm Booster
        * Condition is COMM_BOOSTER_NO_DATA, then maybe a
friendly message to user
        * explaining why their input led to no returned
data. If the Comm Booster
        * Condition is COMM_BOOSTER_ERROR then maybe the
user should be told to log
        * off and try the application at a later time or
call technical support.
        *
        * @param             event
        * Data Object Event to be processed.
        * @exception         CSppageException

```

```

* Tells the caller to discontinue processing
* @exception          CSpSkipException
* Tells the caller to skip processing
*/
protected void processDataObjectEvent
(CSpDataObjectEvent event) throws CSpSkipException, CSpStopException
{
    super.processDataObjectEvent(event);
    switch(event.getId())
    {
        //
        // It appears that
ON_DATABASE_ERROR_EVENT is called before
        //
ON_AFTER_EXECUTE_EVENT. In fact, if the result of the
        //
ON_DATABASE_ERROR_EVENT is STOP, then the ON_AFTER_EXECUTE_EVENT
        // is never called.
We will set the proper COMM BOOSTER condition
        // for
ON_DATABASE_ERROR_EVENT.
        //
case
CSpDataObjectEvent.ON_DATABASE_ERROR_EVENT :
{
    if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG, "ON_DATABASE_ERROR_EVENT");

CSpDBResultStatus the_result_status = event.getDBResults
().getResultStatus();

CSpLog.send(this,CSpLog.WARNING, ""CBethIMSProcedure Result Status =
" +
the_result_status);
int
the_error_code = the_result_status.getErrorCode();
if(28
== the_error_code)
{

CSpLog.send(this,CSpLog.WARNING, "Precise IMS Service Not
Available");

setCommBoosterCondition(COMM_BOOSTER_ERROR);
}
else
if(5 == the_error_code)
{

String the_error_message = the_result_status.getErrorMessage();
if( (the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX)) &&
(the_error_message.endsWith(COMM_BOOSTER_RC_83)))
{
CSpLog.send(this,CSpLog.WARNING, "Comm Booster Return Code Type 83
Found");

setCommBoosterCondition(COMM BOOSTER ERROR);
}
}

```

```
}

else if( (the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX))
&&

(the_error_message.endsWith(COMM_BOOSTER_RC_49)))

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Type 49
Found"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else if( (the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX))
&&

(the_error_message.endsWith(COMM_BOOSTER_RC_5)))

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Type 5
Found"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else if(the_error_message.equals(COMM_BOOSTER_RC_5_MESSAGE))

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Type 5
Found"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else if(the_error_message.startsWith(COMM_BOOSTER_MESSAGE_PREFIX))

{

CSpLog.send(this,CSpLog.WARNING,""Comm Booster Return Code Recieved
But Unknown"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);

}

else

{

CSpLog.send(this,CSpLog.WARNING,""No Condition Mapping for Error
Message ("" +
the_error_message + "")"");

setCommBoosterCondition(COMM_BOOSTER_UNKNOWN);

}

}

else
```

```

        {

CSpLog.send(this,CSpLog.WARNING,""No Condition Mapping for Error Code
"" +
the_error_code);

setCommBoosterCondition(COMM_BOOSTER_UNKNOWN);
}

break;
}
case
CSpDataObjectEvent.ON_BEFORE_EXECUTE_EVENT :
{
if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""ON_BEFORE_EXECUTE_EVENT");
// //
clear the condition before execution
//

resetCommBoosterCondition();

// //
Upon invalid parameters, the procedure is not executed and
// the event flow continues (SKIP). The developer may check that
// the CommBoosterCondition == COMM_BOOSTER_NO_ACTION as well as
// the Invalid Parameter FLAG. ON_AFTER_EXECUTE_EVENT is never
// reached.
//

//juan
if(!
areParametersValid())
{

CSpLog.send(this,CSpLog.WARNING,""Procedure has invalid parameters,
will SKIP"");

throw new CSpSkipException("Procedure has invalid parameters");
}

break;
}
case
CSpDataObjectEvent.ON_AFTER_EXECUTE_EVENT :
{
// //
This is not called if ON_DATABASE_ERROR_EVENT occurs, except
// (see below)
// if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""ON_AFTER_EXECUTE_EVENT");
//
}

```

```

        //
What happens here is that ON_BEFORE_EXECUTE_EVENT resets the      //
the CommBoosterCondition to NO_ACTION.  checkForNoData() will      //
set the CommBoosterCondition to SUCCESS or NO_DATA which can be    //
used by developer          //
                                //
                                if
(getCommBoosterCondition() == COMM_BOOSTER_NO_ACTION)

//
// Just in case somebody sneaks in more event handling and
// we get to this event from a proceeding ON_AFTER_EXECUTE_EVENT
// or ON_DATABASE_ERROR_EVENT, we don't want to checkForNoData to
// avoid ArrayIndexOutOfBoundsException
//
checkForNoData();

if
(LOCAL_DEBUG)
CSpLog.send(this,CSpLog.USER_DEBUG,""Comm Booster Condition = "" +
getCommBoosterCondition());

break;
}
default :
{
CSpLog.send(this,CSpLog.WARNING,""Unknown Data Object Event: "" +
event);
}

}

/*
 * Used internally to determine the
COMM_BOOSTER_NO_DATA condition.
 * There is a limitation with the IMS PAC caused by
the raw interface of the
 * IMS Message Queue. Precise Comm/Booster cannot
distinguish between an
 * IMS transactions which really abends and one which
returns NO DATA.
 * If an IMS transaction (with transaction property
RESPONSE_MODE) doesn't
 * insert a properly sized buffer to the I/O PCB then
Comm/Booster considers
 * this an abend. Therefore, in order to propagate a
NO DATA conditon,
 * <b>All IMS Transactions used with Precise PAC and
this class</b> must
 * return a single record with all SPACES.
 */
protected void checkForNoData()
{

```

```

        //
        // The Multi-Segment stored
procedures for Precise IMS return a single
                                // CSpDBProcedureResult which
encapsulates a CSpDBProcedureParam which
                                // encapsulates a vector of
TNdValue's. When multiple rows are returned,
                                // there will still only be one (1)
TNdValue for each column; that is, a
                                // procedure which returns eight (8)
columns of data will have eight (8)
                                // TNdValue's in the vector
regardless of the number of rows. The
                                // TNdValue's are surely
""constructed types"". There is an interface to
                                // extract both TNdValue and CSpValue
Java types from the CSpDBProcedure-
                                // Param. We will stick with the
CSpValue since TNdValue is more advanced.
                                //
                                CSpDBProcedureResult the_result =
(CSpDBProcedureResult) getLastResults();
                                CSpDBProcedureParam
the_result_parameters = the_result.getReturnParameters();
                                int the_parameter_count =
the_result_parameters.getNumParameters();
                                if(LOCAL_DEBUG)
                                CSpLog.send
(this,CSpLog.USER_DEBUG,""The Number of Parameters = "" +
the_parameter_count);

try
{
                                CSpVector the_vector
= (CSpVector) the_result_parameters.getParameter(0);
                                if(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,""The first vector size = "" +
the_vector.size());
                                if(the_vector.size()
< 1)
                                {
                                //
                                //
should never be reached
                                //
                                }

CSpLog.send(this,CSpLog.ERROR,
""IMS Transaction Design Pattern Failure - Successful Execution
always return at least one row"");

setCommBoosterCondition(COMM_BOOSTER_ERROR);
                                }
                                else if
(the_vector.size() > 1)
                                {
                                if
(LOCAL_DEBUG)

CSpLog.send(this,CSpLog.USER_DEBUG,
""More than one (1) row returned, probably good data"");

```



```

        */
        public static final String COMM_BOOSTER_RC_49 = ""
49"";
        /**
         * String for matching error conditions in
Comm/Booster.
        */
        public static final String COMM_BOOSTER_RC_5 = ""
5"";
        /**
         * String for matching error conditions in
Comm/Booster.
        */
        public static final String COMM_BOOSTER_RC_5_MESSAGE
= ""Comm/Booster rc = 5"";

        /**
         * Unknown Condition for Comm/Booster.
        */
        public static final int COMM_BOOSTER_UNKNOWN = -1;
        /**
         * Initial condition for Comm/Booster before
execution.
        */
        public static final int COMM_BOOSTER_NO_ACTION = 0;
        /**
         * The default condition for Comm/Booster when no
error is found after execution.
        */
        public static final int COMM_BOOSTER_SUCCESS = 1;
        /**
         * Value representing a system failure in the IMS PAC
system. This includes
         * MVS connectivity problems, abend of Precise TCP/IP
relay on MVS, major
         * error in IMS PAC or Comm/Booster middleware
binaries.
        */
        public static final int COMM_BOOSTER_ERROR = 2;
        /**
         * Value representing a condition where an IMS
Transaction was successfully executed but no data was returned.
        */
        public static final int COMM_BOOSTER_NO_DATA = 3;

        /**
         * State of Comm/Booster.
        */
        transient protected int _CommBoosterCondition =
COMM_BOOSTER_NO_ACTION;
        /**
         * State of the input parameters after customized
validation and pre-processing.
        */
        transient protected boolean _ValidProcedureParameters
= true;

        //
        // Comm Booster Condition may be accessed publicly
        //
        /**
         * Access the condition of the Data Object with
respect to Comm/Booster behavior.
         * Most useful after Data Object Execution.
         * @return COMM_BOOSTER_NO_ACTION,
COMM_BOOSTER_SUCCESS, COMM_BOOSTER_ERROR, COMM_BOOSTER_NO_DATA

```

```

        */
    public int getCommBoosterCondition()
    {
        return _CommBoosterCondition;
    }

    /**
     * Checks the state of the input parameters.
     * @return true if any individual input parameter
vetoed the Data Object execution
     */
    public boolean areParametersValid()
    {
        return _ValidProcedureParameters;
    }

    /**
     * Used internally to reset the input parameter
validity flag.
     */
    protected void resetParametersValidFlag()
    {
        _ValidProcedureParameters = true;
    }

    /**
     * Used internally to vote NO to Data Object
execution.
     */
    protected void invalidateParameters()
    {
        _ValidProcedureParameters = false;
    }

    /**
     * Used internally to reset the Comm/Booster
condition before Data Object execution.
     */
    protected void resetCommBoosterCondition()
    {
        _CommBoosterCondition =
COMM_BOOSTER_NO_ACTION;
    }

    /**
     * Used internally to set the Comm/Booster condition
during or after Data Object execution.
     */
    protected void setCommBoosterCondition(int
pCondition)
    {
        if( (pCondition != COMM_BOOSTER_SUCCESS) &&
(pCondition != COMM_BOOSTER_ERROR) &&
(COMM_BOOSTER_UNKNOWN) &&
(pCondition != COMM_BOOSTER_NO_DATA) )
        {
            CSpLog.send
(this,CSpLog.ERROR,""Unknown Comm Booster Condition -> "" +
pCondition);
            _CommBoosterCondition
= COMM_BOOSTER_UNKNOWN;
        }
    }
}

```

```

        }
        _CommBoosterCondition = pCondition;
        if(LOCAL_DEBUG)
        {
            String the_message =
            ""Comm Booster Condition of CBethIMSProcedure is "";
            switch(pCondition)
            {
                case
                COMM_BOOSTER_SUCCESS :
                {
                    the_message = the_message + ""COMM_BOOSTER_SUCCESS"";

                    break;
                }
                case
                COMM_BOOSTER_ERROR :
                {
                    the_message = the_message + ""COMM_BOOSTER_ERROR"";

                    break;
                }
                case
                COMM_BOOSTER_NO_DATA :
                {
                    the_message = the_message + ""COMM_BOOSTER_NO_DATA"";

                    break;
                }
                case
                COMM_BOOSTER_UNKNOWN :
                {
                    the_message = the_message + ""COMM_BOOSTER_UNKNOWN"";

                    break;
                }
            }
            CSpLog.send
            (this,CSpLog.USER_DEBUG,the_message);
        }
    }
}

```

vii.

```

Credit.java
package CreditApp;

import java.io.*;
import java.text.*;
import spider.*;
import spider.database.*;
import java.util.*;
import spider.visual.*;
import spider.session.*;
import spider.util.*;

//[[SPIDER_CLASS BEGIN
public class Credit implements Serializable
{
    private String pastCust = new String();
    private String pastYear = new String();
}

```

```
private String pastName = new String();
private String plate = new String();
private String semiFinished = new String();
private String pipe = new String();
private String sheet = new String();
private String rail = new String();
private String reclamation = new String();
private String other = new String();
private String tinPlate = new String();
private String bar = new String();
private String secondary = new String();
private String otherSpec = new String();
private String businessDivision = new String();
private String domestic = new String();
private String custCode = new String();
private String district = new String();
private String product = new String();
private String cpnyName = new String();
private String incorporatedName = new String();
private String cpnyDivisionName = new String();
private String pOName = new String();
private String headquarterAddress = new String();
private String headquarterBoxNum = new String();
private String headquarterCity = new String();
private String headquarterZip = new String();
private String headquarterForeignAddress = new String();
private String headquarterPhone = new String();
private String headquarterFax = new String();
private String headquarterWeb = new String();
private String purchaseThroughBranch = new String();
private String branchName = new String();
private String branchDivision = new String();
private String branchStAddress = new String();
private String branchCity = new String();
private String branchZip = new String();
private String branchForeignAddress = new String();
private String branchPhone = new String();
private String branchFax = new String();
private String invoiceCorrespondence = new String();
private String otherStreet = new String();
private String otherBoxNum = new String();
private String otherCity = new String();
private String otherZip = new String();
private String contactWithBeth = new String();
private String contactBethName = new String();
private String headquarterState = new String();
private String invoiceForeignAddress = new String();
private String branchState = new String();
private String otherState = new String();
private String apSystem = new String();
private String apMailedTo = new String();
private String buyersName = new String();
private String buyersPhoneNum = new String();
private String buyersEmail = new String();
private String apSupervisorName = new String();
private String apPhoneNum = new String();
private String apEmail = new String();
private String financialSupportName = new String();
private String dun1 = new String();
private String dun2 = new String();
private String dun3 = new String();
private String fedID1 = new String();
private String fedID2 = new String();
private String bankName = new String();
private String bankStreet = new String();
private String bankCity = new String();
```

```
private String bankZip = new String();
private String acctOfficerName = new String();
private String acctOfficerPhoneNum = new String();
private String bankState = new String();
private String bankOther = new String();
private String tradeRefName1 = new String();
private String tradeRefName2 = new String();
private String tradeRefStreet2 = new String();
private String tradeRefCityStZip2 = new String();
private String tradeRefAttn2 = new String();
private String tradeRefPhone2 = new String();
private String tradeRefFax2 = new String();
private String tradeRefName3 = new String();
private String tradeRefStreet3 = new String();
private String tradeRefCityStZip3 = new String();
private String tradeRefAttn3 = new String();
private String tradeRefPhone3 = new String();
private String tradeRefFax3 = new String();
private String tradeRefStreet1 = new String();
private String tradeRefCityStZip1 = new String();
private String tradeRefAttn1 = new String();
private String tradeRefPhone1 = new String();
private String tradeRefFax1 = new String();
private String incorporateName = new String();
private String headStAdd = new String();
private String headquCity = new String();
private String headState = new String();
private String headZip = new String();
private String headquPh = new String();
private String headquFax = new String();
private String headquWw = new String();
private String duns1 = new String();
private String duns2 = new String();
private String duns3 = new String();
private String fed1 = new String();
private String fed2 = new String();
private String bankName1 = new String();
private String bankStr = new String();
private String bankCity1 = new String();
private String bankSt1 = new String();
private String bankZip1 = new String();
private String acctNm = new String();
private String acctPhone = new String();
private String refName1 = new String();
private String refStreet1 = new String();
private String refCity1 = new String();
private String attnName1 = new String();
private String refPh1 = new String();
private String refFaxNum1 = new String();
private String refName2 = new String();
private String refStreet2 = new String();
private String refCity2 = new String();
private String attnName2 = new String();
private String refPh2 = new String();
private String refFaxNum2 = new String();
private String refName3 = new String();
private String refStreet3 = new String();
private String refCity3 = new String();
private String attnName3 = new String();
private String refPh3 = new String();
private String refFaxNum3 = new String();
private String cinNum = new String();
private String cpnyStreet = new String();
private String cpnyCity = new String();
private String custName = new String();
private String custTitle = new String();
```

```
private String cpnyPhone = new String();
private String cpnyState = new String();
private String cpnyZip = new String();
private String cpnyCountry = new String();
private String cpnyOther = new String();
private String dn1 = new String();
private String dn2 = new String();
private String dn3 = new String();
private String fdID1 = new String();
private String fdID2 = new String();
private String cstCde = new String();
private String busDivision = new String();
private String cINNNumber = new String();
private String customerCode = new String();
private String custCode1 = new String();
private String custCode2 = new String();
private String paymentDays = new String();
private String payDays = new String();

//this method creates instance and checks if null to return a new
object or the current one
public static Credit getInstance()
{
    Credit the_Credit = (Credit)
CSpider.getAnyUserSessionObject("uso_Credit");

    if(the_Credit != null)
        return the_Credit;

    else
    {
        return new Credit();
    }
}

public static void saveInstance(Credit pCredit)
{
    CSpider.putAnyUserSessionObject ("uso_Credit", pCredit);
}

public String getPastCust()
{
    return pastCust;
}

public void setPastCust(String pPastCust)
{
    pastCust = pPastCust;
}

public String getPastYear()
{
    return pastYear;
}

public void setPastYear(String pPastYear)
{
    pastYear = pPastYear;
}
```

```
public String getPastName()
{
    return pastName;
}

public void setPastName(String pPastName)
{
    pastName = pPastName;
}

public String getPlate()
{
    return plate;
}

public void setPlate(String pPlate)
{
    plate = pPlate;
}

public String getSemiFinished()
{
    return semiFinished;
}

public void setSemiFinished(String pSemiFinished)
{
    semiFinished = pSemiFinished;
}

public String getPipe()
{
    return pipe;
}

public void setPipe(String pPipe)
{
    pipe = pPipe;
}

public String getSheet()
{
    return sheet;
}

public void setSheet(String pSheet)
{
    sheet = pSheet;
}

public String getRail()
{
    return rail;
}

public void setRail(String pRail)
{
    rail = pRail;
}

public String getReclamation()
{
    return reclamation;
}

public void setReclamation(String pReclamation)
```

```
{  
    reclamation = pReclamation;  
}  
  
public String getOther()  
{  
    return other;  
}  
  
public void setOther(String pOther)  
{  
    other = pOther;  
}  
  
public String getTinPlate()  
{  
    return tinPlate;  
}  
  
public void setTinPlate(String pTinPlate)  
{  
    tinPlate = pTinPlate;  
}  
  
public String getBar()  
{  
    return bar;  
}  
  
public void setBar(String pBar)  
{  
    bar = pBar;  
}  
  
public String getSecondary()  
{  
    return secondary;  
}  
  
public void setSecondary(String pSecondary)  
{  
    secondary = pSecondary;  
}  
  
public String getOtherSpec()  
{  
    return otherSpec;  
}  
  
public void setOtherSpec(String pOtherSpec)  
{  
    otherSpec = pOtherSpec;  
}  
  
public String getBusinessDivision()  
{  
    return businessDivision;  
}  
  
public void setBusinessDivision(String pBusinessDivision)  
{  
    businessDivision = pBusinessDivision;  
}  
  
public String getDomestic()  
{
```

```
    return domestic;
}

public void setDomestic(String pDomestic)
{
    domestic = pDomestic;
}

public String getCINNum()
{
    return cinNum;
}

public void setCINNum(String pCINNum)
{
    cinNum = pCINNum;
}

public String getCustCode()
{
    return custCode;
}

public void setCustCode(String pCustCode)
{
    custCode = pCustCode;
}

public String getDistrict()
{
    return district;
}

public void setDistrict(String pDistrict)
{
    district = pDistrict;
}

public String getProduct()
{
    return product;
}

public void setProduct(String pProduct)
{
    product = pProduct;
}

public String getCpnyName()
{
    return cpnyName;
}

public void setCpnyName(String pCpnyName)
{
    cpnyName = pCpnyName;
}

public String getIncorporatedName()
{
    return incorporatedName;
}

public void setIncorporatedName(String pIncorporatedName)
```

```
{  
    incorporatedName = pIncorporatedName;  
}  
  
public String getCpnyDivisionName()  
{  
    return cpnyDivisionName;  
}  
  
public void setCpnyDivisionName(String pCpnyDivisionName)  
{  
    cpnyDivisionName = pCpnyDivisionName;  
}  
  
public String getPOName()  
{  
    return pOName;  
}  
  
public void setPOName(String pPOName)  
{  
    pOName = pPOName;  
}  
  
public String getHeadquarterAddress()  
{  
    return headquarterAddress;  
}  
  
public void setHeadquarterAddress(String pHeadquarterAddress)  
{  
    headquarterAddress = pHeadquarterAddress;  
}  
  
public String getHeadquarterBoxNum()  
{  
    return headquarterBoxNum;  
}  
  
public void setHeadquarterBoxNum(String pHeadquarterBoxNum)  
{  
    headquarterBoxNum = pHeadquarterBoxNum;  
}  
  
public String getHeadquarterCity()  
{  
    return headquarterCity;  
}  
  
public void setHeadquarterCity(String pHeadquarterCity)  
{  
    headquarterCity = pHeadquarterCity;  
}  
  
public String getHeadquarterZip()  
{  
    return headquarterZip;  
}  
  
public void setHeadquarterZip(String pHeadquarterZip)  
{  
    headquarterZip = pHeadquarterZip;  
}  
  
public String getHeadquarterForeignAddress()  
{
```

```
    return headquarterForeignAddress;
}

public void setHeadquarterForeignAddress(String pHeadquarterForeignAddress)
{
    headquarterForeignAddress = pHeadquarterForeignAddress;
}

public String getHeadquarterPhone()
{
    return headquarterPhone;
}

public void setHeadquarterPhone(String pHeadquarterPhone)
{
    headquarterPhone = pHeadquarterPhone;
}

public String getHeadquarterFax()
{
    return headquarterFax;
}

public void setHeadquarterFax(String pHeadquarterFax)
{
    headquarterFax = pHeadquarterFax;
}

public String getHeadquarterWeb()
{
    return headquarterWeb;
}

public void setHeadquarterWeb(String pHeadquarterWeb)
{
    headquarterWeb = pHeadquarterWeb;
}

public String getPurchaseThroughBranch()
{
    return purchaseThroughBranch;
}

public void setPurchaseThroughBranch(String pPurchaseThroughBranch)
{
    purchaseThroughBranch = pPurchaseThroughBranch;
}

public String getBranchName()
{
    return branchName;
}

public void setBranchName(String pBranchName)
{
    branchName = pBranchName;
}

public String getBranchDivision()
{
    return branchDivision;
}

public void setBranchDivision(String pBranchDivision)
{
```

```
    branchDivision = pBranchDivision;
}

public String getBranchStAddress()
{
    return branchStAddress;
}

public void setBranchStAddress(String pBranchStAddress)
{
    branchStAddress = pBranchStAddress;
}

public String getBranchCity()
{
    return branchCity;
}

public void setBranchCity(String pBranchCity)
{
    branchCity = pBranchCity;
}

public String getBranchZip()
{
    return branchZip;
}

public void setBranchZip(String pBranchZip)
{
    branchZip = pBranchZip;
}

public String getBranchForeignAddress()
{
    return branchForeignAddress;
}

public void setBranchForeignAddress(String pBranchForeignAddress)
{
    branchForeignAddress = pBranchForeignAddress;
}

public String getBranchPhone()
{
    return branchPhone;
}

public void setBranchPhone(String pBranchPhone)
{
    branchPhone = pBranchPhone;
}

public String getBranchFax()
{
    return branchFax;
}

public void setBranchFax(String pBranchFax)
{
    branchFax = pBranchFax;
}

public String getInvoiceCorrespondence()
{
    return invoiceCorrespondence;
}
```

```
}

public void setInvoiceCorrespondence(String pInvoiceCorrespondence)
{
    invoiceCorrespondence = pInvoiceCorrespondence;
}

public String getOtherStreet()
{
    return otherStreet;
}

public void setOtherStreet(String pOtherStreet)
{
    otherStreet = pOtherStreet;
}

public String getOtherBoxNum()
{
    return otherBoxNum;
}

public void setOtherBoxNum(String pOtherBoxNum)
{
    otherBoxNum = pOtherBoxNum;
}

public String getOtherCity()
{
    return otherCity;
}

public void setOtherCity(String pOtherCity)
{
    otherCity = pOtherCity;
}

public String getOtherZip()
{
    return otherZip;
}

public void setOtherZip(String pOtherZip)
{
    otherZip = pOtherZip;
}

public String getContactWithBeth()
{
    return contactWithBeth;
}

public void setContactWithBeth(String pContactWithBeth)
{
    contactWithBeth = pContactWithBeth;
}

public String getContactBethName()
{
    return contactBethName;
}

public void setContactBethName(String pContactBethName)
{
    contactBethName = pContactBethName;
}
```

```
public String getHeadquarterState()
{
    return headquarterState;
}

public void setHeadquarterState(String pHeadquarterState)
{
    headquarterState = pHeadquarterState;
}

public String getInvoiceForeignAddress()
{
    return invoiceForeignAddress;
}

public void setInvoiceForeignAddress(String pInvoiceForeignAddress)
{
    invoiceForeignAddress = pInvoiceForeignAddress;
}

public String getBranchState()
{
    return branchState;
}

public void setBranchState(String pBranchState)
{
    branchState = pBranchState;
}

public String getOtherState()
{
    return otherState;
}

public void setOtherState(String pOtherState)
{
    otherState = pOtherState;
}

public String getApSystem()
{
    return apSystem;
}

public void setApSystem(String pApSystem)
{
    apSystem = pApSystem;
}

public String getApMailedTo()
{
    return apMailedTo;
}

public void setApMailedTo(String pApMailedTo)
{
    apMailedTo = pApMailedTo;
}

public String getBuyersName()
{
    return buyersName;
}
```

```
public void setBuyersName(String pBuyersName)
{
    buyersName = pBuyersName;
}

public String getBuyersPhoneNum()
{
    return buyersPhoneNum;
}

public void setBuyersPhoneNum(String pBuyersPhoneNum)
{
    buyersPhoneNum = pBuyersPhoneNum;
}

public String getBuyersEmail()
{
    return buyersEmail;
}

public void setBuyersEmail(String pBuyersEmail)
{
    buyersEmail = pBuyersEmail;
}

public String getApSupervisorName()
{
    return apSupervisorName;
}

public void setApSupervisorName(String pApSupervisorName)
{
    apSupervisorName = pApSupervisorName;
}

public String getApPhoneNum()
{
    return apPhoneNum;
}

public void setApPhoneNum(String pApPhoneNum)
{
    apPhoneNum = pApPhoneNum;
}

public String getApEmail()
{
    return apEmail;
}

public void setApEmail(String pApEmail)
{
    apEmail = pApEmail;
}

public String getFinancialSupportName()
{
    return financialSupportName;
}

public void setFinancialSupportName(String pFinancialSupportName)
{
    financialSupportName = pFinancialSupportName;
}

public String getDun1()
```

```
{  
    return dun1;  
}  
  
public void setDun1(String pDun1)  
{  
    dun1 = pDun1;  
}  
  
public String getDun2()  
{  
    return dun2;  
}  
  
public void setDun2(String pDun2)  
{  
    dun2 = pDun2;  
}  
  
public String getDun3()  
{  
    return dun3;  
}  
  
public void setDun3(String pDun3)  
{  
    dun3 = pDun3;  
}  
  
public String getFedID1()  
{  
    return fedID1;  
}  
  
public void setFedID1(String pFedID1)  
{  
    fedID1 = pFedID1;  
}  
  
public String getFedID2()  
{  
    return fedID2;  
}  
  
public void setFedID2(String pFedID2)  
{  
    fedID2 = pFedID2;  
}  
  
public String getBankName()  
{  
    return bankName;  
}  
  
public void setBankName(String pBankName)  
{  
    bankName = pBankName;  
}  
  
public String getBankStreet()  
{  
    return bankStreet;  
}  
  
public void setBankStreet(String pBankStreet)  
{
```

```
    bankStreet = pBankStreet;
}

public String getBankCity()
{
    return bankCity;
}

public void setBankCity(String pBankCity)
{
    bankCity = pBankCity;
}

public String getBankZip()
{
    return bankZip;
}

public void setBankZip(String pBankZip)
{
    bankZip = pBankZip;
}

public String getAcctOfficerName()
{
    return acctOfficerName;
}

public void setAcctOfficerName(String pAcctOfficerName)
{
    acctOfficerName = pAcctOfficerName;
}

public String getAcctOfficerPhoneNum()
{
    return acctOfficerPhoneNum;
}

public void setAcctOfficerPhoneNum(String pAcctOfficerPhoneNum)
{
    acctOfficerPhoneNum = pAcctOfficerPhoneNum;
}

public String getBankState()
{
    return bankState;
}

public void setBankState(String pBankState)
{
    bankState = pBankState;
}

public String getBankOther()
{
    return bankOther;
}

public void setBankOther(String pBankOther)
{
    bankOther = pBankOther;
}

public String getTradeRefName1()
{
    return tradeRefName1;
```

```
}

public void setTradeRefName1(String pTradeRefName1)
{
    tradeRefName1 = pTradeRefName1;
}

public String getTradeRefName2()
{
    return tradeRefName2;
}

public void setTradeRefName2(String pTradeRefName2)
{
    tradeRefName2 = pTradeRefName2;
}

public String getTradeRefStreet2()
{
    return tradeRefStreet2;
}

public void setTradeRefStreet2(String pTradeRefStreet2)
{
    tradeRefStreet2 = pTradeRefStreet2;
}

public String getTradeRefCityStZip2()
{
    return tradeRefCityStZip2;
}

public void setTradeRefCityStZip2(String pTradeRefCityStZip2)
{
    tradeRefCityStZip2 = pTradeRefCityStZip2;
}

public String getTradeRefAttn2()
{
    return tradeRefAttn2;
}

public void setTradeRefAttn2(String pTradeRefAttn2)
{
    tradeRefAttn2 = pTradeRefAttn2;
}

public String getTradeRefPhone2()
{
    return tradeRefPhone2;
}

public void setTradeRefPhone2(String pTradeRefPhone2)
{
    tradeRefPhone2 = pTradeRefPhone2;
}

public String getTradeRefFax2()
{
    return tradeRefFax2;
}

public void setTradeRefFax2(String pTradeRefFax2)
{
    tradeRefFax2 = pTradeRefFax2;
}
```

```
public String getTradeRefName3()
{
    return tradeRefName3;
}

public void setTradeRefName3(String pTradeRefName3)
{
    tradeRefName3 = pTradeRefName3;
}

public String getTradeRefStreet3()
{
    return tradeRefStreet3;
}

public void setTradeRefStreet3(String pTradeRefStreet3)
{
    tradeRefStreet3 = pTradeRefStreet3;
}

public String getTradeRefCityStZip3()
{
    return tradeRefCityStZip3;
}

public void setTradeRefCityStZip3(String pTradeRefCityStZip3)
{
    tradeRefCityStZip3 = pTradeRefCityStZip3;
}

public String getTradeRefAttn3()
{
    return tradeRefAttn3;
}

public void setTradeRefAttn3(String pTradeRefAttn3)
{
    tradeRefAttn3 = pTradeRefAttn3;
}

public String getTradeRefPhone3()
{
    return tradeRefPhone3;
}

public void setTradeRefPhone3(String pTradeRefPhone3)
{
    tradeRefPhone3 = pTradeRefPhone3;
}

public String getTradeRefFax3()
{
    return tradeRefFax3;
}

public void setTradeRefFax3(String pTradeRefFax3)
{
    tradeRefFax3 = pTradeRefFax3;
}

public String getTradeRefStreet1()
{
    return tradeRefStreet1;
}
```

```
public void setTradeRefStreet1(String pTradeRefStreet1)
{
    tradeRefStreet1 = pTradeRefStreet1;
}

public String getTradeRefCityStZip1()
{
    return tradeRefCityStZip1;
}

public void setTradeRefCityStZip1(String pTradeRefCityStZip1)
{
    tradeRefCityStZip1 = pTradeRefCityStZip1;
}

public String getTradeRefAttn1()
{
    return tradeRefAttn1;
}

public void setTradeRefAttn1(String pTradeRefAttn1)
{
    tradeRefAttn1 = pTradeRefAttn1;
}

public String getTradeRefPhone1()
{
    return tradeRefPhone1;
}

public void setTradeRefPhone1(String pTradeRefPhone1)
{
    tradeRefPhone1 = pTradeRefPhone1;
}

public String getTradeRefFax1()
{
    return tradeRefFax1;
}

public void setTradeRefFax1(String pTradeRefFax1)
{
    tradeRefFax1 = pTradeRefFax1;
}

public boolean isValid()
{
    if(getIncorporatedName().trim().length() == 0)
    {
        return false;
    }
    if(getHeadquarterAddress().trim().length() == 0)
    {
        return false;
    }
    if(getHeadquarterCity().trim().length() == 0)
    {
        return false;
    }
    if(getHeadquarterState().trim().length() == 0)
    {
        return false;
    }
    if(getHeadquarterZip().trim().length() == 0)
    {
```

```
    return false;
}

if(getDun1().trim().length() == 0)
{
    return false;
}
if(getDun2().trim().length() == 0)
{
    return false;
}
if(getDun3().trim().length() == 0)
{
    return false;
}
if(getFedID1().trim().length() == 0)
{
    return false;
}
if(getFedID2().trim().length() == 0)
{
    return false;
}
if(getBankName().trim().length() == 0)
{
    return false;
}
if(getBankStreet().trim().length() == 0)
{
    return false;
}
if(getBankCity().trim().length() == 0)
{
    return false;
}
if(getBankZip().trim().length() == 0)
{
    return false;
}

if(getTradeRefName1().trim().length() == 0)
{
    return false;
}

if(getTradeRefCityStZip1().trim().length() == 0)
{
    return false;
}

if(getTradeRefName2().trim().length() == 0)
{
    return false;
}

if(getTradeRefCityStZip2().trim().length() == 0)
{
    return false;
}

if(getTradeRefName3().trim().length() == 0)
{
    return false;
}

if(getTradeRefCityStZip3().trim().length() == 0)
```

```
{  
    return false;  
}  
else  
    return true;  
}  
  
public String getIncorporateName()  
{  
    return incorporateName;  
}  
  
public void setIncorporateName(String pIncorporateName)  
{  
    incorporateName = pIncorporateName;  
}  
  
public String getHeadStAdd()  
{  
    return headStAdd;  
}  
  
public void setHeadStAdd(String pHeadStAdd)  
{  
    headStAdd = pHeadStAdd;  
}  
  
public String getHeadquCity()  
{  
    return headquCity;  
}  
  
public void setHeadquCity(String pHeadquCity)  
{  
    headquCity = pHeadquCity;  
}  
  
public String getHeadState()  
{  
    return headState;  
}  
  
public void setHeadState(String pHeadState)  
{  
    headState = pHeadState;  
}  
  
public String getHeadZip()  
{  
    return headZip;  
}  
  
public void setHeadZip(String pHeadZip)  
{  
    headZip = pHeadZip;  
}  
  
public String getHeadquPh()  
{  
    return headquPh;  
}  
  
public void setHeadquPh(String pHeadquPh)  
{  
    headquPh = pHeadquPh;  
}
```

```
}

public String getHeadquFax()
{
    return headquFax;
}

public void setHeadquFax(String pHeadquFax)
{
    headquFax = pHeadquFax;
}

public String getHeadquWWW()
{
    return headquWWW;
}

public void setHeadquWWW(String pHeadquWWW)
{
    headquWWW = pHeadquWWW;
}

public String getDuns1()
{
    return duns1;
}

public void setDuns1(String pDuns1)
{
    duns1 = pDuns1;
}

public String getDuns2()
{
    return duns2;
}

public void setDuns2(String pDuns2)
{
    duns2 = pDuns2;
}

public String getDuns3()
{
    return duns3;
}

public void setDuns3(String pDuns3)
{
    duns3 = pDuns3;
}

public String getFed1()
{
    return fed1;
}

public void setFed1(String pFed1)
{
    fed1 = pFed1;
}

public String getFed2()
{
    return fed2;
}
```

```
public void setFed2(String pFed2)
{
    fed2 = pFed2;
}

public String getBankName1()
{
    return bankName1;
}

public void setBankName1(String pBankName1)
{
    bankName1 = pBankName1;
}

public String getBankStr()
{
    return bankStr;
}

public void setBankStr(String pBankStr)
{
    bankStr = pBankStr;
}

public String getBankCity1()
{
    return bankCity1;
}

public void setBankCity1(String pBankCity1)
{
    bankCity1 = pBankCity1;
}

public String getBankSt1()
{
    return bankSt1;
}

public void setBankSt1(String pBankSt1)
{
    bankSt1 = pBankSt1;
}

public String getBankZip1()
{
    return bankZip1;
}

public void setBankZip1(String pBankZip1)
{
    bankZip1 = pBankZip1;
}

public String getAcctNm()
{
    return acctNm;
}

public void setAcctNm(String pAcctNm)
{
    acctNm = pAcctNm;
}
```

```
public String getAcctPhone()
{
    return acctPhone;
}

public void setAcctPhone(String pAcctPhone)
{
    acctPhone = pAcctPhone;
}

public String getRefName1()
{
    return refName1;
}

public void setRefName1(String pRefName1)
{
    refName1 = pRefName1;
}

public String getRefStreet1()
{
    return refStreet1;
}

public void setRefStreet1(String pRefStreet1)
{
    refStreet1 = pRefStreet1;
}

public String getRefCity1()
{
    return refCity1;
}

public void setRefCity1(String pRefCity1)
{
    refCity1 = pRefCity1;
}

public String getAttnName1()
{
    return attnName1;
}

public void setAttnName1(String pAttnName1)
{
    attnName1 = pAttnName1;
}

public String getRefPh1()
{
    return refPh1;
}

public void setRefPh1(String pRefPh1)
{
    refPh1 = pRefPh1;
}

public String getRefFaxNum1()
{
    return refFaxNum1;
}
```

```
public void setRefFaxNum1(String pRefFaxNum1)
{
    refFaxNum1 = pRefFaxNum1;
}

public String getRefName2()
{
    return refName2;
}

public void setRefName2(String pRefName2)
{
    refName1 = pRefName2;
}

public String getRefStreet2()
{
    return refStreet2;
}

public void setRefStreet2(String pRefStreet2)
{
    refStreet2 = pRefStreet2;
}

public String getRefCity2()
{
    return refCity2;
}

public void setRefCity2(String pRefCity2)
{
    refCity2 = pRefCity2;
}

public String getAttnName2()
{
    return attnName2;
}

public void setAttnName2(String pAttnName2)
{
    attnName2 = pAttnName2;
}

public String getRefPh2()
{
    return refPh2;
}

public void setRefPh2(String pRefPh2)
{
    refPh2 = pRefPh2;
}

public String getRefFaxNum2()
{
    return refFaxNum2;
}

public void setRefFaxNum2(String pRefFaxNum2)
{
    refFaxNum2 = pRefFaxNum2;
}

public String getRefName3()
```

```
{  
    return refName3;  
}  
  
public void setRefName3(String pRefName3)  
{  
    refName3 = pRefName3;  
}  
  
public String getRefStreet3()  
{  
    return refStreet3;  
}  
  
public void setRefStreet3(String pRefStreet3)  
{  
    refStreet3 = pRefStreet3;  
}  
  
public String getRefCity3()  
{  
    return refCity3;  
}  
  
public void setRefCity3(String pRefCity3)  
{  
    refCity3 = pRefCity3;  
}  
  
    public String getAttnName3()  
{  
    return attnName3;  
}  
  
public void setAttnName3(String pAttnName3)  
{  
    attnName3 = pAttnName3;  
}  
  
public String getRefPh3()  
{  
    return refPh3;  
}  
  
public void setRefPh3(String pRefPh3)  
{  
    refPh3 = pRefPh3;  
}  
  
public String getRefFaxNum3()  
{  
    return refFaxNum3;  
}  
  
public void setRefFaxNum3(String pRefFaxNum3)  
{  
    refFaxNum3 = pRefFaxNum3;  
}  
  
public String getCpnyStreet()  
{  
    return cpnyStreet;  
}  
  
public void setCpnyStreet(String pCpnyStreet)  
{
```

```
    cpnyStreet = pCpnyStreet;
}

public String getCpnyCity()
{
    return cpnyCity;
}

public void setCpnyCity(String pCpnyCity)
{
    cpnyCity = pCpnyCity;
}

public String getCustName()
{
    return custName;
}

public void setCustName(String pCustName)
{
    custName = pCustName;
}

public String getCustTitle()
{
    return custTitle;
}

public void setCustTitle(String pCustTitle)
{
    custTitle = pCustTitle;
}

public String getCpnyPhone()
{
    return cpnyPhone;
}

public void setCpnyPhone(String pCpnyPhone)
{
    cpnyPhone = pCpnyPhone;
}

public String getCpnyState()
{
    return cpnyState;
}

public void setCpnyState(String pCpnyState)
{
    cpnyState = pCpnyState;
}

public String getCpnyZip()
{
    return cpnyZip;
}

public void setCpnyZip(String pCpnyZip)
{
    cpnyZip = pCpnyZip;
}

public String getCpnyCountry()
{
    return cpnyCountry;
}
```

```
}

public void setCpnyCountry(String pCpnyCountry)
{
    cpnyCountry = pCpnyCountry;
}

public String getCpnyOther()
{
    return cpnyOther;
}

public void setCpnyOther(String pCpnyOther)
{
    cpnyOther = pCpnyOther;
}

public String getDn1()
{
    return dn1;
}

public void setDn1(String pDn1)
{
    dn1 = pDn1;
}

public String getDn2()
{
    return dn2;
}

public void setDn2(String pDn2)
{
    dn2 = pDn2;
}

public String getDn3()
{
    return dn3;
}

public void setDn3(String pDn3)
{
    dn3 = pDn3;
}

public String getFdID1()
{
    return fdID1;
}

public void setFdID1(String pFdID1)
{
    fdID1 = pFdID1;
}

public String getFdID2()
{
    return fdID2;
}

public void setFdID2(String pFdID2)
{
    fdID2 = pFdID2;
}
```

```
public String getCstCde()
{
    return cstCde;
}

public void setCstCde(String pCstCde)
{
    cstCde = pCstCde;
}

public String getBusDivision()
{
    return busDivision;
}

public void setBusDivision(String pBusDivision)
{
    busDivision = pBusDivision;
}

public String getCIINumber()
{
    return cIINumber;
}

public void setCIINumber(String pCIINumber)
{
    cIINumber = pCIINumber;
}

public String getCustomerCode()
{
    return customerCode;
}

public void setCustomerCode(String pCustomerCode)
{
    customerCode = pCustomerCode;
}

public String getCustCode1()
{
    return custCode1;
}

public void setCustCode1(String pCustCode1)
{
    custCode1 = pCustCode1;
}

public String getCustCode2()
{
    return custCode2;
}

public void setCustCode2(String pCustCode2)
{
    custCode2 = pCustCode2;
}

public String getPaymentDays()
{
    return paymentDays;
}
```

```

public void setPaymentDays(String pPaymentDays)
{
    paymentDays = pPaymentDays;
}

public String getPayDays()
{
    return payDays;
}

public void setPayDays(String pPayDays)
{
    payDays = pPayDays;
}
}

```

viii.

CreditApp.java
// This file has been generated by NetDynamics Studio on Thu Apr. 27,
2000 08:46

```

package CreditApp;

import spider.*;
import spider.event.*;
import spider.visual.*;
import spider.session.*;
import spider.util.*;

```

```

//[[SPIDER_CLASS BEGIN
public class CreditApp extends spider.CSpProject
//]]SPIDER_CLASS END
{
    //[[SPIDER_EVENTS BEGIN
    //]]SPIDER_EVENTS END
}

```

ix.

CreditApp2.java
// This file has been generated by NetDynamics Studio on Wed May. 03,
2000 09:08

```

package CreditApp;

import java.awt.event.*;
import spider.event.*;
import spider.database.*;
import spider.visual.*;
import spider.util.*;
import spider.*;
import spider.session.*;
import java.lang.*;

```

```

//[[SPIDER_CLASS BEGIN
public class CreditApp2 extends spider.visual.CSpPage
//]]SPIDER_CLASS END
{
    //the credit object is the object the fields are
    being set and get
    private transient Credit the_CreditObject;
    //[[SPIDER_EVENTS BEGIN
    //[[SPIDER_EVENT<this_onBeforeLoadEvent>

```

```

        public int this_onBeforeLoadEvent(CSpPageEvent event)
    {
        the_CreditObject = (Credit)
Credit.getInstance();
        return (PROCEED);
    }
//]]SPIDER_EVENT<this_onBeforeLoadEvent>

        //[[SPIDER_EVENT
<IncorporatedName_onBeforeDisplayEvent>
        public int IncorporatedName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_incorporatedName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_incorporatedName.setValue(new
CSpString(the_CreditObject.getIncorporatedName()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<IncorporatedName_onBeforeDisplayEvent>

        //[[SPIDER_EVENT
<CpnyDivisionName_onBeforeDisplayEvent>
        public int CpnyDivisionName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_cpnyDivisionName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_cpnyDivisionName.setValue(new
CSpString(the_CreditObject.getCpnyDivisionName()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<CpnyDivisionName_onBeforeDisplayEvent> .

        //[[SPIDER_EVENT<POName_onBeforeDisplayEvent>
        public int POName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_pOName = (CSpTextBox)
event.getSource();
    if(the_CreditObject != null)
        the_pOName.setValue(new CSpString
(the_CreditObject.getPOName()));
    return (PROCEED);
}
//]]SPIDER_EVENT<POName_onBeforeDisplayEvent>

        //[[SPIDER_EVENT
<HeadquarterAddress_onBeforeDisplayEvent>
        public int HeadquarterAddress_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_headquarterAddress =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterAddress.setValue(new
CSpString(the_CreditObject.getHeadquarterAddress()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<HeadquarterAddress_onBeforeDisplayEvent>

```

```

        //[[SPIDER_EVENT
<HeadquarterBoxNum_onBeforeDisplayEvent>
    public int HeadquarterBoxNum_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_headquarterBoxNum =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterBoxNum.setValue(new
CSpString(the_CreditObject.getHeadquarterBoxNum()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<HeadquarterBoxNum_onBeforeDisplayEvent>

        //[[SPIDER_EVENT
<HeadquarterCity_onBeforeDisplayEvent>
    public int HeadquarterCity_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_headquarterCity =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterCity.setValue(new
CSpString(the_CreditObject.getHeadquarterCity()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<HeadquarterCity_onBeforeDisplayEvent>

        //[[SPIDER_EVENT<HeadquarterZip_onBeforeDisplayEvent>
    public int HeadquarterZip_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_headquarterZip =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterZip.setValue(new
CSpString(the_CreditObject.getHeadquarterZip()));
    return (PROCEED);
}
//]]SPIDER_EVENT<HeadquarterZip_onBeforeDisplayEvent>

        //[[SPIDER_EVENT
<HeadquarterForeignAddress_onBeforeDisplayEvent>
    public int
HeadquarterForeignAddress_onBeforeDisplayEvent(CSpDisplayEvent event)
{
    CSpTextBox
the_headquarterForeignAddress = (CSpTextBox) event.getSource();
    if(the_CreditObject != null)

the_headquarterForeignAddress.setValue(new CSpString
(the_CreditObject.getHeadquarterForeignAddress()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<HeadquarterForeignAddress_onBeforeDisplayEvent>

        //[[SPIDER_EVENT
<HeadquarterPhone_onBeforeDisplayEvent>
    public int HeadquarterPhone_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_headquarterPhone =

```

```

(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterPhone.setValue(new
CSpString(the_CreditObject.getHeadquarterPhone()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<HeadquarterPhone_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquarterFax_onBeforeDisplayEvent>
public int HeadquarterFax_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_headquarterFax =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterFax.setValue(new
CSpString(the_CreditObject.getHeadquarterFax()));
    return (PROCEED);
}
//]]SPIDER_EVENT<HeadquarterFax_onBeforeDisplayEvent>

//[[SPIDER_EVENT<HeadquarterWeb_onBeforeDisplayEvent>
public int HeadquarterWeb_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_headquarterWeb =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterWeb.setValue(new
CSpString(the_CreditObject.getHeadquarterWeb()));
    return (PROCEED);
}
//]]SPIDER_EVENT<HeadquarterWeb_onBeforeDisplayEvent>

//[[SPIDER_EVENT
<PurchaseThroughBranch_onBeforeDisplayEvent>
public int PurchaseThroughBranch_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpRadioButtonsGroup
the_purchaseThroughBranch = (CSpRadioButtonsGroup) event.getSource();
    if(the_CreditObject != null)
        the_purchaseThroughBranch.setValue
(new CSpString(the_CreditObject.getPurchaseThroughBranch()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<PurchaseThroughBranch_onBeforeDisplayEvent>

//[[SPIDER_EVENT<BranchName_onBeforeDisplayEvent>
public int BranchName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_branchName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_branchName.setValue(new
CSpString(the_CreditObject.getBranchName()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BranchName_onBeforeDisplayEvent>

//[[SPIDER_EVENT<BranchDivision_onBeforeDisplayEvent>
public int BranchDivision_onBeforeDisplayEvent
(CSpDisplayEvent event)

```

```

        {
            CSpTextBox the_branchDivision =
(CSpTextBox) event.getSource();
            if(the_CreditObject != null)
                the_branchDivision.setValue(new
CSpString(the_CreditObject.getBranchDivision()));
            return (PROCEED);
        }
    //]]SPIDER_EVENT<BranchDivision_onBeforeDisplayEvent>

    //[[SPIDER_EVENT
<BranchStAddress_onBeforeDisplayEvent>
        public int BranchStAddress_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_branchStAddress =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_branchStAddress.setValue(new
CSpString(the_CreditObject.getBranchStAddress()));
    return (PROCEED);
}
    //]]SPIDER_EVENT
<BranchStAddress_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BranchCity_onBeforeDisplayEvent>
        public int BranchCity_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_branchCity =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_branchCity.setValue(new
CSpString(the_CreditObject.getBranchCity()));
    return (PROCEED);
}
    //]]SPIDER_EVENT<BranchCity_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BranchZip_onBeforeDisplayEvent>
        public int BranchZip_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_branchZip =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_branchZip.setValue(new
CSpString(the_CreditObject.getBranchZip()));
    return (PROCEED);
}
    //]]SPIDER_EVENT<BranchZip_onBeforeDisplayEvent>

    //[[SPIDER_EVENT
<BranchForeignAddress_onBeforeDisplayEvent>
        public int BranchForeignAddress_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_branchForeignAddress =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_branchForeignAddress.setValue
(new CSpString(the_CreditObject.getBranchForeignAddress()));
    return (PROCEED);
}
    //]]SPIDER_EVENT
<BranchForeignAddress_onBeforeDisplayEvent>

```

```

    //[[SPIDER_EVENT<BranchPhone_onBeforeDisplayEvent>
    public int BranchPhone_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_branchPhone =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_branchPhone.setValue(new
CSpString(the_CreditObject.getBranchPhone()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BranchPhone_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BranchFax_onBeforeDisplayEvent>
    public int BranchFax_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_branchFax =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_branchFax.setValue(new
CSpString(the_CreditObject.getBranchFax()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BranchFax_onBeforeDisplayEvent>

    //[[SPIDER_EVENT
<InvoiceCorrespondence_onBeforeDisplayEvent>
    public int InvoiceCorrespondence_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpRadioButtonsGroup
the_invoiceCorrespondence = (CSpRadioButtonsGroup) event.getSource();
    if(the_CreditObject != null)
        the_invoiceCorrespondence.setValue
(new CSpString(the_CreditObject.getInvoiceCorrespondence()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<InvoiceCorrespondence_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<OtherStreet_onBeforeDisplayEvent>
    public int OtherStreet_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_otherStreet =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_otherStreet.setValue(new
CSpString(the_CreditObject.getOtherStreet()));
    return (PROCEED);
}
//]]SPIDER_EVENT<OtherStreet_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<OtherBoxNum_onBeforeDisplayEvent>
    public int OtherBoxNum_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_otherBoxNum =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_otherBoxNum.setValue(new
CSpString(the_CreditObject.getOtherBoxNum()));
    return (PROCEED);
}
//]]SPIDER_EVENT<OtherBoxNum_onBeforeDisplayEvent>

```

```

//[[SPIDER_EVENT<OtherCity_onBeforeDisplayEvent>
public int OtherCity_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_otherCity =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_otherCity.setValue(new
CSpString(the_CreditObject.getOtherCity()));
    return (PROCEED);
}
//]]SPIDER_EVENT<OtherCity_onBeforeDisplayEvent>

//[[SPIDER_EVENT<OtherZip_onBeforeDisplayEvent>
public int OtherZip_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_otherZip =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_otherZip.setValue(new CSpString
(the_CreditObject.getOtherZip()));
    return (PROCEED);
}
//]]SPIDER_EVENT<OtherZip_onBeforeDisplayEvent>

//[[SPIDER_EVENT
<ContactWithBeth_onBeforeDisplayEvent>
public int ContactWithBeth_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpRadioButtonsGroup
the_contactWithBeth = (CSpRadioButtonsGroup) event.getSource();
    if(the_CreditObject != null)
        the_contactWithBeth.setValue(new
CSpString(the_CreditObject.getContactWithBeth()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<ContactWithBeth_onBeforeDisplayEvent>

//[[SPIDER_EVENT
<ContactBethName_onBeforeDisplayEvent>
public int ContactBethName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_contactBethName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_contactBethName.setValue(new
CSpString(the_CreditObject.getContactBethName()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<ContactBethName_onBeforeDisplayEvent>

//[[SPIDER_EVENT
<HeadquarterState_onBeforeDisplayEvent>
public int HeadquarterState_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpComboBox the_headquarterState =
(CSpComboBox) event.getSource();
    if(the_CreditObject != null)
        the_headquarterState.setValue(new

```

```

CSpString(the_CreditObject.getHeadquarterState()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<HeadquarterState_onBeforeDisplayEvent>

    //[[SPIDER_EVENT
<InvoiceForeignAddress_onBeforeDisplayEvent>
    public int InvoiceForeignAddress_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_invoiceForeignAddress
= (CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_invoiceForeignAddress.setValue
(new CSpString(the_CreditObject.getInvoiceForeignAddress()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<InvoiceForeignAddress_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<BranchState_onBeforeDisplayEvent>
    public int BranchState_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpComboBox the_branchState =
(CSpComboBox) event.getSource();
    if(the_CreditObject != null)
        the_branchState.setValue(new
CSpString(the_CreditObject.getBranchState()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BranchState_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<OtherState_onBeforeDisplayEvent>
    public int OtherState_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpComboBox the_otherState =
(CSpComboBox) event.getSource();
    if(the_CreditObject != null)
        the_otherState.setValue(new
CSpString(the_CreditObject.getOtherState()));
    return (PROCEED);
}
//]]SPIDER_EVENT<OtherState_onBeforeDisplayEvent>

    //[[SPIDER_EVENT<GotoApp1_onWebEvent>
    public int GotoApp1_onWebEvent(CSpWebEvent event)
{
    the_CreditObject = Credit.getInstance
();
    /* setting all the fields on this
page so when the user hits the ""prev"" button
        their information is saved */
    the_CreditObject.setIncorporatedName
(getDisplayField("IncorporatedName")).getValue().toString());
    the_CreditObject.setCpnyDivisionName(getDisplayField
("CpnyDivisionName")).getValue().toString());
    the_CreditObject.setPOName
(getDisplayField("POName")).getValue().toString());
    the_CreditObject.setHeadquarterAddress(getDisplayField
("HeadquarterAddress")).getValue().toString());
    the_CreditObject.setHeadquarterBoxNum
(getDisplayField("HeadquarterBoxNum")).getValue().toString());
}

```

```

        the_CreditObject.setHeadquarterCity
(getDisplayField("HeadquarterCity").getValue().toString());
        the_CreditObject.setHeadquarterZip
(getDisplayField("HeadquarterZip").getValue().toString());

the_CreditObject.setHeadquarterForeignAddress (getDisplayField
("HeadquarterForeignAddress").getValue().toString());
        the_CreditObject.setHeadquarterPhone
(getDisplayField("HeadquarterPhone").getValue().toString());
        the_CreditObject.setHeadquarterFax
(getDisplayField("HeadquarterFax").getValue().toString());
        the_CreditObject.setHeadquarterWeb
(getDisplayField("HeadquarterWeb").getValue().toString());

the_CreditObject.setPurchaseThroughBranch (getDisplayField
("PurchaseThroughBranch").getValue().toString());
        the_CreditObject.setBranchName
(getDisplayField("BranchName").getValue().toString());
        the_CreditObject.setBranchDivision
(getDisplayField("BranchDivision").getValue().toString());
        the_CreditObject.setBranchStAddress
(getDisplayField("BranchStAddress").getValue().toString());
        the_CreditObject.setBranchCity
(getDisplayField("BranchCity").getValue().toString());
        the_CreditObject.setBranchZip
(getDisplayField("BranchZip").getValue().toString());
        the_CreditObject.setBranchForeignAddress (getDisplayField
("BranchForeignAddress").getValue().toString());
        the_CreditObject.setBranchPhone
(getDisplayField("BranchPhone").getValue().toString());
        the_CreditObject.setBranchFax
(getDisplayField("BranchFax").getValue().toString());

the_CreditObject.setInvoiceCorrespondence (getDisplayField
("InvoiceCorrespondence").getValue().toString());
        the_CreditObject.setOtherStreet
(getDisplayField("OtherStreet").getValue().toString());
        the_CreditObject.setOtherBoxNum
(getDisplayField("OtherBoxNum").getValue().toString());
        the_CreditObject.setOtherCity
(getDisplayField("OtherCity").getValue().toString());
        the_CreditObject.setOtherZip
(getDisplayField("OtherZip").getValue().toString());
        the_CreditObject.setContactWithBeth
(getDisplayField("ContactWithBeth").getValue().toString());
        the_CreditObject.setContactBethName
(getDisplayField("ContactBethName").getValue().toString());
        the_CreditObject.setHeadquarterState
(getDisplayField("HeadquarterState").getValue().toString());

the_CreditObject.setInvoiceForeignAddress (getDisplayField
("InvoiceForeignAddress").getValue().toString());
        the_CreditObject.setBranchState
(getDisplayField("BranchState").getValue().toString());
        the_CreditObject.setOtherState
(getDisplayField("OtherState").getValue().toString());
        the_CreditObject.setBranchCity
(getDisplayField("BranchCity").getValue().toString());

        Credit.getInstance
(the_CreditObject);
        return load("CreditApplication");
    }
//]]SPIDER_EVENT<GotoAppl_onWebEvent>

```

```

// [[SPIDER_EVENT<GotoApp3_onWebEvent>
public int GotoApp3_onWebEvent(CSpWebEvent event)
{
    Credit the_CreditObject =
Credit.getInstance();

        /* setting all the fields on this
page so when the user hits the ""next"" button
their information is saved */
the_CreditObject.setIncorporatedName
getDisplayField("IncorporatedName").getValue().toString());
the_CreditObject.setCpnyDivisionName(getDisplayField
("CpnyDivisionName").getValue().toString());
the_CreditObject.setPOName
getDisplayField("POName").getValue().toString());

the_CreditObject.setHeadquarterAddress(getDisplayField
("HeadquarterAddress").getValue().toString());
the_CreditObject.setHeadquarterBoxNum
getDisplayField("HeadquarterBoxNum").getValue().toString());
the_CreditObject.setHeadquarterCity
getDisplayField("HeadquarterCity").getValue().toString());
the_CreditObject.setHeadquarterZip
getDisplayField("HeadquarterZip").getValue().toString());

the_CreditObject.setHeadquarterForeignAddress(getDisplayField
("HeadquarterForeignAddress").getValue().toString());
the_CreditObject.setHeadquarterPhone
getDisplayField("HeadquarterPhone").getValue().toString());
the_CreditObject.setHeadquarterFax
getDisplayField("HeadquarterFax").getValue().toString());
the_CreditObject.setHeadquarterWeb
getDisplayField("HeadquarterWeb").getValue().toString());

the_CreditObject.setPurchaseThroughBranch(getDisplayField
("PurchaseThroughBranch").getValue().toString());
the_CreditObject.setBranchName
getDisplayField("BranchName").getValue().toString());
the_CreditObject.setBranchDivision
getDisplayField("BranchDivision").getValue().toString());
the_CreditObject.setBranchStAddress
getDisplayField("BranchStAddress").getValue().toString());
the_CreditObject.setBranchCity
getDisplayField("BranchCity").getValue().toString());
the_CreditObject.setBranchZip
getDisplayField("BranchZip").getValue().toString());
the_CreditObject.setBranchForeignAddress(getDisplayField
("BranchForeignAddress").getValue().toString());
the_CreditObject.setBranchPhone
getDisplayField("BranchPhone").getValue().toString());
the_CreditObject.setBranchFax
getDisplayField("BranchFax").getValue().toString());

the_CreditObject.setInvoiceCorrespondence(getDisplayField
("InvoiceCorrespondence").getValue().toString());
the_CreditObject.setOtherStreet
getDisplayField("OtherStreet").getValue().toString());
the_CreditObject.setOtherBoxNum
getDisplayField("OtherBoxNum").getValue().toString());
the_CreditObject.setOtherCity
getDisplayField("OtherCity").getValue().toString());
the_CreditObject.setOtherZip
getDisplayField("OtherZip").getValue().toString());
the_CreditObject.setContactWithBeth
getDisplayField("ContactWithBeth").getValue().toString());
the_CreditObject.setContactBethName

```

```

(getDisplayField("ContactBethName").getValue().toString());
                           the_CreditObject.setHeadquarterState
(getDisplayField("HeadquarterState").getValue().toString());

the_CreditObject.setInvoiceForeignAddress(getDisplayField
("InvoiceForeignAddress").getValue().toString());
                           the_CreditObject.setBranchState
(getDisplayField("BranchState").getValue().toString());
                           the_CreditObject.setOtherState
(getDisplayField("OtherState").getValue().toString());
                           the_CreditObject.setBranchCity
(getDisplayField("BranchCity").getValue().toString());

                           Credit.saveInstance
(the_CreditObject);
                           return load("CreditApp3");
}
//]]SPIDER_EVENT<GotoApp3_onWebEvent>

//[[SPIDER_EVENT<GoToCreditHmPg_onWebEvent>
public int GoToCreditHmPg_onWebEvent(CSpWebEvent
event)
{
                           return load("CreditDeptHomePg");
}
//]]SPIDER_EVENT<GoToCreditHmPg_onWebEvent>

//]]SPIDER_EVENTS END
}

x.

CreditApp3.java
// This file has been generated by NetDynamics Studio on Thu May. 04,
2000 10:23

package CreditApp;

import java.awt.event.*;
import spider.event.*;
import spider.database.*;
import spider.visual.*;
import spider.util.*;
import spider.*;
import spider.session.*;
import java.lang.*;

//[[SPIDER_CLASS BEGIN
public class CreditApp3 extends spider.visual.CSpPage
//]]SPIDER_CLASS END
{
    private transient Credit the_CreditObject;
    //[[SPIDER_EVENTS BEGIN

    //[[SPIDER_EVENT<this_onBeforeLoadEvent>
    public int this_onBeforeLoadEvent(CSpPageEvent event)
    {
        the_CreditObject = (Credit)
Credit.getInstance();
        return (PROCEED);
    }
    //]]SPIDER_EVENT<this_onBeforeLoadEvent>

    //[[SPIDER_EVENT<EmailBankInfo_onWebEvent>
    public int EmailBankInfo_onWebEvent(CSpWebEvent

```

```

event)
{
    int command = PROCEED;
    command = doAction(event);
    return(command);
}
//]]SPIDER_EVENT<EmailBankInfo_onWebEvent>

//[[SPIDER_EVENT<BuyersName_onBeforeDisplayEvent>
public int BuyersName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_buyersName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_buyersName.setValue(new
CSpString(the_CreditObject.getBuyersName()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BuyersName_onBeforeDisplayEvent>

//[[SPIDER_EVENT<BuyersPhoneNum_onBeforeDisplayEvent>
public int BuyersPhoneNum_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_buyersPhoneNum =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_buyersPhoneNum.setValue(new
CSpString(the_CreditObject.getBuyersPhoneNum()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BuyersPhoneNum_onBeforeDisplayEvent>

//[[SPIDER_EVENT<BuyersEmail_onBeforeDisplayEvent>
public int BuyersEmail_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_buyersEmail =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_buyersEmail.setValue(new
CSpString(the_CreditObject.getBuyersEmail()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BuyersEmail_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Dun1_onBeforeDisplayEvent>
public int Dun1_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    CSpTextBox the_dun1 = (CSpTextBox)
event.getSource();
    if(the_CreditObject != null)
        the_dun1.setValue(new CSpString
(the_CreditObject.getDun1()));
    return (PROCEED);
}
//]]SPIDER_EVENT<Dun1_onBeforeDisplayEvent>

//[[SPIDER_EVENT<Dun2_onBeforeDisplayEvent>
public int Dun2_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    CSpTextBox the_dun2 = (CSpTextBox)
event.getSource();

```

```

                if(the_CreditObject != null)
                    the_dun2.setValue(new CSpString
(the_CreditObject.getDun2()));
                return (PROCEED);
            }
        //]]SPIDER_EVENT<Dun2_onBeforeDisplayEvent>
        //[[SPIDER_EVENT<Dun3_onBeforeDisplayEvent>
public int Dun3_onBeforeDisplayEvent(CSpDisplayEvent
event)
{
    CSpTextBox the_dun3 = (CSpTextBox)
event.getSource();
    if(the_CreditObject != null)
        the_dun3.setValue(new CSpString
(the_CreditObject.getDun3()));
    return (PROCEED);
}
//]]SPIDER_EVENT<Dun3_onBeforeDisplayEvent>
//[[SPIDER_EVENT<FedID1_onBeforeDisplayEvent>
public int FedID1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_fedID1 = (CSpTextBox)
event.getSource();
    if(the_CreditObject != null)
        the_fedID1.setValue(new CSpString
(the_CreditObject.getFedID1()));
    return (PROCEED);
}
//]]SPIDER_EVENT<FedID1_onBeforeDisplayEvent>
//[[SPIDER_EVENT<FedID2_onBeforeDisplayEvent>
public int FedID2_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_fedID2 = (CSpTextBox)
event.getSource();
    if(the_CreditObject != null)
        the_fedID2.setValue(new CSpString
(the_CreditObject.getFedID2()));
    return (PROCEED);
}
//]]SPIDER_EVENT<FedID2_onBeforeDisplayEvent>
//[[SPIDER_EVENT<BankName_onBeforeDisplayEvent>
public int BankName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_bankName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_bankName.setValue(new CSpString
(the_CreditObject.getBankName()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BankName_onBeforeDisplayEvent>
//[[SPIDER_EVENT<BankStreet_onBeforeDisplayEvent>
public int BankStreet_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_bankStreet =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)

```

```

                the_bankStreet.setValue(new
CSpString(the_CreditObject.getBankStreet()));
                return (PROCEED);
            }
        //]]SPIDER_EVENT<BankStreet_onBeforeDisplayEvent>
        //[[SPIDER_EVENT<BankCity_onBeforeDisplayEvent>
        public int BankCity_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_bankCity =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_bankCity.setValue(new CSpString
(the_CreditObject.getBankCity()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BankCity_onBeforeDisplayEvent>
//[[SPIDER_EVENT<BankZip_onBeforeDisplayEvent>
        public int BankZip_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_bankZip = (CSpTextBox)
event.getSource();
    if(the_CreditObject != null)
        the_bankZip.setValue(new CSpString
(the_CreditObject.getBankZip()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BankZip_onBeforeDisplayEvent>
//[[SPIDER_EVENT
<AcctOfficerName_onBeforeDisplayEvent>
        public int AcctOfficerName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_acctOfficerName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_acctOfficerName.setValue(new
CSpString(the_CreditObject.getAcctOfficerName()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<AcctOfficerName_onBeforeDisplayEvent>
//[[SPIDER_EVENT
<AcctOfficerPhoneNum_onBeforeDisplayEvent>
        public int AcctOfficerPhoneNum_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_acctOfficerPhoneNum =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_acctOfficerPhoneNum.setValue(new
CSpString(the_CreditObject.getAcctOfficerPhoneNum()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<AcctOfficerPhoneNum_onBeforeDisplayEvent>
//[[SPIDER_EVENT<BankState_onBeforeDisplayEvent>
        public int BankState_onBeforeDisplayEvent
(CSpDisplayEvent event)
{

```

```

        CSpComboBox the_bankState =
(CSpComboBox) event.getSource();
        if(the_CreditObject != null)
            the_bankState.setValue(new CSpString
(the_CreditObject.getBankState()));
        return (PROCEED);
    }
//]]SPIDER_EVENT<BankState_onBeforeDisplayEvent>

//[[SPIDER_EVENT<BankOther_onBeforeDisplayEvent>
public int BankOther_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_bankOther =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_bankOther.setValue(new CSpString
(the_CreditObject.getBankOther()));
    return (PROCEED);
}
//]]SPIDER_EVENT<BankOther_onBeforeDisplayEvent>

//[[SPIDER_EVENT<ApSystem_onBeforeDisplayEvent>
public int ApSystem_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpRadioButtonsGroup the_apSystem =
(CSpRadioButtonsGroup) event.getSource();
    if(the_CreditObject != null)
        the_apSystem.setValue(new CSpString
(the_CreditObject.getApSystem()));
    return (PROCEED);
}
//]]SPIDER_EVENT<ApSystem_onBeforeDisplayEvent>

//[[SPIDER_EVENT<ApMailedTo_onBeforeDisplayEvent>
public int ApMailedTo_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpRadioButtonsGroup the_apMailedTo =
(CSpRadioButtonsGroup) event.getSource();
    if(the_CreditObject != null)
        the_apMailedTo.setValue(new
CSpString(the_CreditObject.getApMailedTo()));
    return (PROCEED);
}
//]]SPIDER_EVENT<ApMailedTo_onBeforeDisplayEvent>

//[[SPIDER_EVENT
<ApSupervisorName_onBeforeDisplayEvent>
public int ApSupervisorName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_apSupervisorName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_apSupervisorName.setValue(new
CSpString(the_CreditObject.getApSupervisorName()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<ApSupervisorName_onBeforeDisplayEvent>

//[[SPIDER_EVENT<ApPhoneNum_onBeforeDisplayEvent>
public int ApPhoneNum_onBeforeDisplayEvent
(CSpDisplayEvent event)

```

```

        {
            CSpTextBox the_apPhoneNum =
(CSpTextBox) event.getSource(); if(the_CreditObject != null)
                                the_apPhoneNum.setValue(new
CSpString(the_CreditObject.getApPhoneNum()));
                                return (PROCEED);
        }
//]]SPIDER_EVENT<ApPhoneNum_onBeforeDisplayEvent>

//[[SPIDER_EVENT<ApEmail_onBeforeDisplayEvent>
public int ApEmail_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_apEmail = (CSpTextBox)
event.getSource(); if(the_CreditObject != null)
                                the_apEmail.setValue(new CSpString
(the_CreditObject.getApEmail()));
                                return (PROCEED);
}
//]]SPIDER_EVENT<ApEmail_onBeforeDisplayEvent>

//[[SPIDER_EVENT<GoToCreditHmPg_onWebEvent>
public int GoToCreditHmPg_onWebEvent(CSpWebEvent
event)
{
    return load("//CreditDeptHomePg");
}
//]]SPIDER_EVENT<GoToCreditHmPg_onWebEvent>

//[[SPIDER_EVENT<GoToApp2_onWebEvent>
public int GoToApp2_onWebEvent(CSpWebEvent event)
{
    Credit the_CreditObject =
Credit.getInstance();
/* setting all the fields on this
page so when the user hits the ""prev"" button
their information is saved */
the_CreditObject.setApSystem
(getDisplayField("ApSystem").getValue().toString());
the_CreditObject.setApMailedTo(getDisplayField
("ApMailedTo").getValue().toString());
the_CreditObject.setBuyersName
(getDisplayField("BuyersName").getValue().toString());
the_CreditObject.setBuyersPhoneNum
(getDisplayField("BuyersPhoneNum").getValue().toString());
the_CreditObject.setBuyersEmail
(getDisplayField("BuyersEmail").getValue().toString());
the_CreditObject.setApSupervisorName
(getDisplayField("ApSupervisorName").getValue().toString());
the_CreditObject.setApPhoneNum
(getDisplayField("ApPhoneNum").getValue().toString());
the_CreditObject.setApEmail
(getDisplayField("ApEmail").getValue().toString());

the_CreditObject.setFinancialSupportName(getDisplayField
("FinancialSupportName").getValue().toString());
the_CreditObject.setDun1
(getDisplayField("Dun1").getValue().toString());
the_CreditObject.setDun2
(getDisplayField("Dun2").getValue().toString());
the_CreditObject.setDun3
(getDisplayField("Dun3").getValue().toString());
the_CreditObject.setFedID1
(getDisplayField("FedID1").getValue().toString());

```

```

        the_CreditObject.setFedID2
(getDisplayField("FedID2").getValue().toString());
        the_CreditObject.setBankName
(getDisplayField("BankName").getValue().toString());
        the_CreditObject.setBankStreet
(getDisplayField("BankStreet").getValue().toString());
        the_CreditObject.setBankCity
(getDisplayField("BankCity").getValue().toString());
        the_CreditObject.setBankZip(displayField
("BankZip").getValue().toString());
        the_CreditObject.setAcctOfficerName
(getDisplayField("AcctOfficerName").getValue().toString());

the_CreditObject.setAcctOfficerPhoneNum(displayField
("AcctOfficerPhoneNum").getValue().toString());
        the_CreditObject.setBankState
(getDisplayField("BankState").getValue().toString());
        the_CreditObject.setBankOther
(getDisplayField("BankOther").getValue().toString());

                Credit.saveInstance
(the_CreditObject);
        return load("CreditApp2");
    }
//]]SPIDER_EVENT<GoToApp2_onWebEvent>

//[[SPIDER_EVENT<GoToApp4_onWebEvent>
public int GoToApp4_onWebEvent(CSpWebEvent event)
{
        Credit the_CreditObject =
Credit.getInstance();
        /* setting all the fields on this
page so when the user hits the "next" button
        their information is saved */
        the_CreditObject.setApSystem
(displayField("ApSystem").getValue().toString());
        the_CreditObject.setApMailedTo(displayField
("ApMailedTo").getValue().toString());
        the_CreditObject.setBuyersName
(displayField("BuyersName").getValue().toString());
        the_CreditObject.setBuyersPhoneNum
(displayField("BuyersPhoneNum").getValue().toString());
        the_CreditObject.setBuyersEmail
(displayField("BuyersEmail").getValue().toString());
        the_CreditObject.setApSupervisorName
(displayField("ApSupervisorName").getValue().toString());
        the_CreditObject.setApPhoneNum
(displayField("ApPhoneNum").getValue().toString());
        the_CreditObject.setApEmail
(displayField("ApEmail").getValue().toString());

the_CreditObject.setFinancialSupportName(displayField
("FinancialSupportName").getValue().toString());
        the_CreditObject.setDun1
(displayField("Dun1").getValue().toString());
        the_CreditObject.setDun2
(displayField("Dun2").getValue().toString());
        the_CreditObject.setDun3
(displayField("Dun3").getValue().toString());
        the_CreditObject.setFedID1
(displayField("FedID1").getValue().toString());
        the_CreditObject.setFedID2
(displayField("FedID2").getValue().toString());
        the_CreditObject.setBankName
(displayField("BankName").getValue().toString());
        the_CreditObject.setBankStreet

```

```

(getDisplayField("BankStreet").getValue().toString());
    the_CreditObject.setBankCity
(getDisplayField("BankCity").getValue().toString());
    the_CreditObject.setBankZip(getDisplayField
("BankZip").getValue().toString());
    the_CreditObject.setAcctOfficerName
(getDisplayField("AcctOfficerName").getValue().toString());

the_CreditObject.setAcctOfficerPhoneNum(getDisplayField
("AcctOfficerPhoneNum").getValue().toString());
    the_CreditObject.setBankState
(getDisplayField("BankState").getValue().toString());
    the_CreditObject.setBankOther
(getDisplayField("BankOther").getValue().toString());

                Credit.saveInstance
(the_CreditObject);
return load("CreditApp4");

}

//]]SPIDER_EVENT<GoToApp4_onWebEvent>

//[[SPIDER_EVENT
<FinancialSupportName_onBeforeDisplayEvent>
public int FinancialSupportName_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox the_financialSupportName =
(CSpTextBox) event.getSource();
    if(the_CreditObject != null)
        the_financialSupportName.setValue
(new CSpString(the_CreditObject.getFinancialSupportName()));
    return (PROCEED);
}
//]]SPIDER_EVENT
<FinancialSupportName_onBeforeDisplayEvent>

//]]SPIDER_EVENTS END
}

xi.

CreditApp4.java
// This file has been generated by NetDynamics Studio on Thu May. 04,
2000 16:32

package CreditApp;

import java.awt.event.*;
import spider.event.*;
import spider.database.*;
import spider.visual.*;
import spider.util.*;
import spider.*;
import spider.session.*;
import java.lang.*;

//[[SPIDER_CLASS BEGIN
public class CreditApp4 extends spider.visual.CSpPage
//]]SPIDER_CLASS END
{
    private transient Credit the_CreditObject;

    //[[SPIDER_EVENTS BEGIN
    //[[SPIDER_EVENT<this_onBeforeLoadEvent>

```

```
    public int this_onBeforeLoadEvent(CSpPageEvent event)
    {
        the_CreditObject = (Credit)
Credit.getInstance();
        return(PROCEED);
    }
//]]SPIDER_EVENT<this_onBeforeLoadEvent>

//[[SPIDER_EVENT<TradeRefName1_onBeforeDisplayEvent>
public int TradeRefName1_onBeforeDisplayEvent
(CSpDisplayEvent event)
{
    CSpTextBox" Joe Berenato
/O=LBLW LLC/OU=LBLW/CN=RECIPIENTS/CN=JBERENATO EX Scott
Wakeman /O=LBLW LLC/OU=LBLW/CN=RECIPIENTS/CN=Swakeman EX
```

WHAT IS CLAIMED IS:

1. A method of permitting selective computer access to information about a first entity by a second entity, comprising the steps of:
 - storing predetermined information about a first entity on a storage device of a host computer, wherein access to the storage device is decisively limited;
 - creating via a host entity a general identification specific to the first entity, and through the host computer linking the predetermined information to the general information of the first entity;
 - creating via the first entity a specific identification permitting access to the predetermined information stored on the storage device; and
 - informing via the first identity a second entity of the general and specific identification, and therefore permitting the second entity to access the storage device and review the predetermined information.
- 15 2. The method of claim 1, including the step of storing credit information as the predetermined information.
3. The method of claim 2, including the step of storing credit information of a predetermined duration.
- 20 4. The method of claim 3, including the step of storing credit information created within the previous five years.
5. The method of claim 2, including the step of storing information extracted from said credit information.
- 25 6. The method of claim 1, including the step of permitting access to the storage device via a website of the host entity.
- 30 7. The method of claim 1, including the step of creating the general identification specific to the first entity from the group consisting of alphanumeric sequences, numeric sequences, and combinations thereof.
8. The method of claim 1, including the step of creating the specific

- identification via the first entity from the group consisting of alphanumeric sequences, numeric sequences, and combinations thereof.
9. The method of claim 1, including the step of limiting the general and
5 specific identification to no more than ten digits.
10. The method of claim 1, including the step of storing the specific identification on the storage device of the host entity.
- 10 11. The method of claim 8, including the step of permitting the first entity to assign or change the specific identification.
12. A computer system for storing information concerning transactions between a first entity and a second entity and selectively permitting a third entity to
15 access selected portions of said information comprising:
a data storage device controllable by said second entity and connected to a network accessible by said third entity;
information stored on said data storage device concerning transactions between said first entity and said second entity;
20 a general first party identifier stored on said data storage device and associated with said information;
a specific first party identifier stored on said data storage device and associated with said information; and
a data processor for providing access to a selected portion of said information
25 over said network to a third party after said third party sends said general first party identifier and said specific first party identifier over said network to said data processor.
- 30 13. The system of claim 12, wherein said data storage device comprises a memory device capable of receiving and storing information from at least the first entity and the second entity.
14. The system of claim 12, wherein said general identifier and said specific

identifier each comprise an alphanumerical sequence, a numerical sequence, or combination thereof.

15. The system of claim 14, wherein said general identifier sequence
5 comprises fewer than ten characters.

16. The system of claim 12, wherein said specific first party identifier is changeable by said first party.

10 17. The system of claim 12, wherein said information comprises credit information.

18. The system of claim 17, wherein said credit information comprises credit information from a limited period of time.

15 19. The system of claim 12, wherein said selected portion of said information comprises less than all said information.

20 20. A method of permitting selective computer access to credit information about a first entity by a second entity, comprising the steps of:

storing credit information about a first entity on a storage device of a host computer, wherein access to the storage device is decisively limited;
creating via a host entity a general identification specific to the first entity, and through the host computer linking the credit information to the general identification of the first entity;

25 creating via the first entity a specific identification permitting access to the credit information stored on the storage device; and

informing via the first identity a second entity of the general and specific identification; and

30 permitting the second entity to access the storage device and review the credit information through use of the storage device.

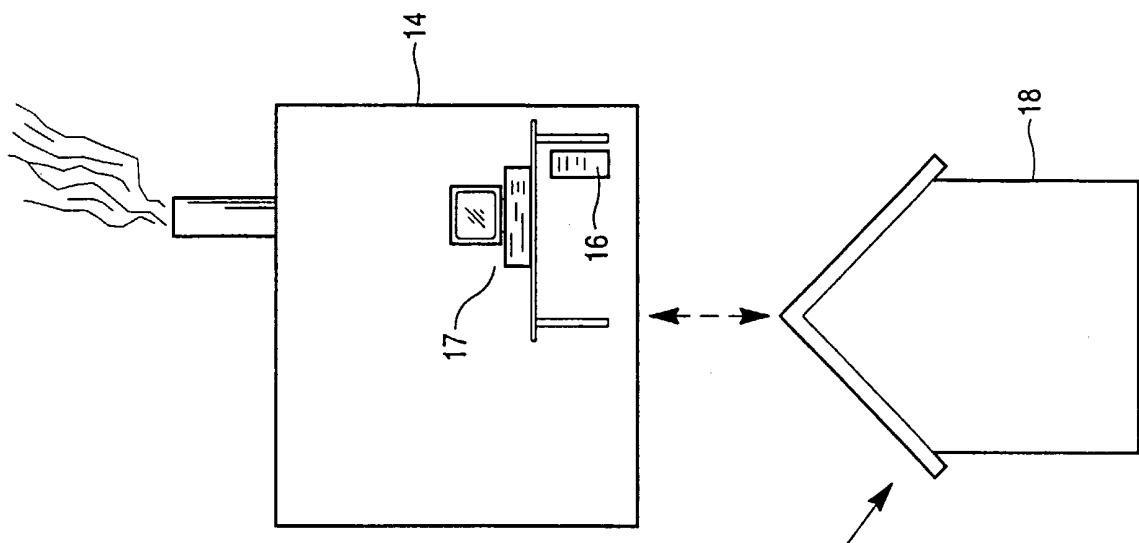


Fig. 1

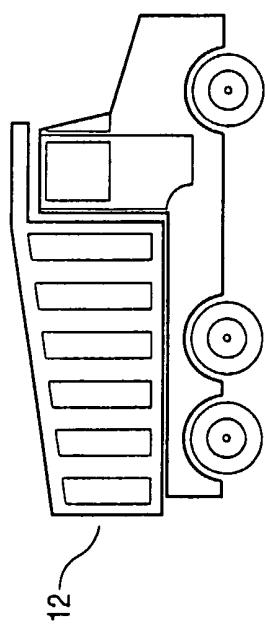


Fig. 2

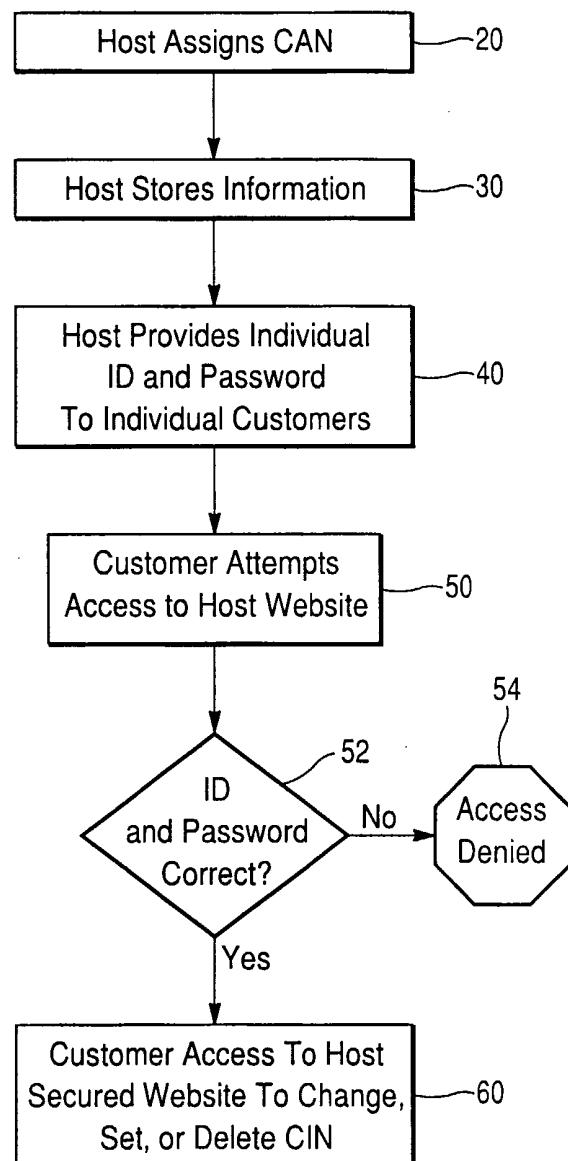


Fig. 3

