

Credit Card Validation Requirements Specification

1. Scope

1.1 Overview

The Credit Card Validation Component provides the algorithms to validate the accuracy and validity of a credit card number. However, this component only validates formatting and does not guarantee the credit card number will be authorized by a bank. The algorithm provides a simple check digit calculation and pattern match algorithm to verify the number is formatted properly.

1.2 Logic Requirements

1.2.1 Supported Credit Cards

The following credit card types will be supported. It will be possible to configure the component to accept only certain types of credit cards. This list will also be configurable; users will be able to add additional card types.

- MasterCard
- VISA
- American Express
- Diners Club/ Carte Blanche
- Discover
- JCB

1.2.2 Prefix Validation

Each credit card type has a specific length and beginning sequence. A valid credit card must match the following table.

Credit Card Type	Prefix	Length of CC Number
MasterCard	51-55	16
Visa	4	13 or 16
American Express	34 or 37	15
Diners Club/ Carte Blanche	300-305 or 36 or 38	14
Discover	6011	16
JCB	3	16
JCB	2131,1800	15

1.3 Required Algorithms

The component must implement the LUHN Formula (Mod 10) algorithm. Furthermore, the component must be able to validate the prefix and length of a number.

1.4 Example of the Software Usage

E-Commerce websites need to validate credit card numbers prior to sending the information

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

None.

2.1.3 *Environment Requirements*

- Development language: Java1.4
- Compile target: Java1.3, Java1.4

2.1.4 *Namespace*

com.topcoder.ecommerce.creditcard.validation

3. **Software Requirements**

3.1 **Administration Requirements**

3.1.1 *What elements of the application need to be configurable?*

- None

3.2 **Technical Constraints**

3.2.1 *Are there particular frameworks or standards that are required?*

None.

3.2.2 *TopCoder Software Component Dependencies:*

None.

****Please review the [TopCoder Software component catalog](#) for existing components that can be used in the design.**

3.2.3 *Third Party Component, Library, or Product Dependencies:*

None.

3.2.4 *QA Environment:*

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003

3.3 **Design Constraints**

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 **Required Documentation**

3.4.1 *Design Documentation*

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification
- Test Plan

3.4.2 *Help / User Documentation*

Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.