> PROGRESS® OPENEDGE® APPLICATIONS IN A PCI-DSS ENVIRONMENT

Michael Jacobs

www.progress.com

PROGRESS software
BUSINESS MAKING PROGRESS™

TABLE OF CONTENTS

PAYMENT CARD INDUSTRY–DATA SECURITY STANDARD

INTRODUCTION

If your Progress® OpenEdge® application handles credit card information or runs in a credit card data environment[1], it will have to comply with the Data Security Standard (DSS) published by the Payment Card Industry Security Standards Council[2] (PCI council). Becoming PCI-DSS compliant always raises two questions:

1. What is PCI-DSS compliance and what does it mean for my OpenEdge application?

2. How do I use OpenEdge to become PCI-DSS-compliant?

This paper will provide a brief overview of the PCI-DSS and its impact on an OpenEdge payment card application[3]. It does not seek to interpret or define what the PCI-DSS is. Rather, it provides information about what OpenEdge offers that can be used in creating a PCI-DSS-compliant payment card application. It will help if you have some level of understanding of what the PCI-DSS is and the security processes upon which it is based.

PCI-DSS STANDARD OVERVIEW

The PCI Council's mission is to reduce credit card fraud by reducing the ability of intruders to use common computing security flaws to steal credit card data. To remove those common security flaws the PCI Council created the Data Security Standard (DSS), which protects credit card data throughout its entire lifecycle. The DSS is a collection of well-known security technologies, processes, and best practices that are proven to reduce electronic data theft when they are applied correctly.

What Is PCI-DSS Compliance?

The PCI Council requires every merchant, card processor, and card service provider (hereafter referred to as "merchants") to comply with the DSS

[1] *A credit card data environment is defined by the PCI Council as having direct access to network traffic, OS processes, or physical disk storage that handles credit card data*

[2] *Payment Card Industry (PCI) primary members are VISA, MasterCard, American Express, Discover, and JCB.*

[3] *A payment card application is a software application that is involved in handling credit card or card holder information.*

if they wish to do credit card business with the PCI Council's members. The PCI Council holds those merchants fully responsible for ensuring that all in-scope [hardware and software] systems[4] comply with the DSS requirements.

To become, and remain compliant, a merchant's in-scope systems must undergo a yearly audit by the PCI Council. So that every system a merchant runs does not have to be PCI-DSS-compliant, the PCI Council introduced the definition of scope. PCI scope limits which business systems have to be PCI-DSS-compliant by isolating them behind a DSS-compliant firewall on the merchant's internal network.

For distributed OpenEdge applications, this means that some of its components, such as its clients, may run outside the firewall and not have to be DSS-compliant. Other components such as servers and databases, especially those that may handle credit card data, will run inside the firewall and must be DSS-compliant. The PCI Council's definition of scope may also impact your OpenEdge application even if it does not handle credit card data. If a merchant has installed your application behind the firewall where it has direct access to card data applications and storage, this means that it must be DSS-compliant.

The PCI Council has divided DSS certification process into four tiers, in which each tier is defined by how many credit card transactions a merchant handled each year. The certification process involves auditing a merchant's network and either submitting to a self-assessment questionnaire (SAQ) or having a qualified security assessor (QSA) do a physical audit. If a merchant suffers a data break-in, regardless of which compliance tier it is in, it can be required to undergo a full QSA audit at the PCI's discretion. Your in-scope OpenEdge application will be audited according to the tier a merchant falls into.

INSIDE THE PCI-DSS

The DSS standard is divided into twelve general requirements that incorporate aspects of network, server, application, data, process, and physical security. Some of the requirements will be provided by the merchants themselves, such as process and physical plant security. Some

[4] *A PCI-DSS system is defined as any network, OS, application, or storage provider that comes in contact with, or could be used to come into contact with, credit card data.*

will be provided by network and OS providers. The bulk of the requirements are application and data storage related, and it will be your OpenEdge application's responsibility to meet them. Your application will need to address network data encryption, application security, data storage security, user authentication, data access control, and auditing.

Periodically, the PCI Council will release new versions of the DSS. The new versions address gray areas in the standard, security requirements for new computing technologies, and counters to new technologies and methods of stealing electronic data. So becoming PCI-DSS-compliant is not a one-time event. Your OpenEdge application will be required to meet the requirements of new versions as they are released. The PCI Council announces when new versions will be released and the amount of time that merchants will have to incorporate the upgraded systems into their operations. You do not want to be caught unaware and try to meet new requirements at the last minute.

WHERE TO FIND PCI-DSS INFORMATION

The road to success in creating a PCI-DSS-compliant payment card application starts with understanding the definition of compliance and what the auditors will determine is an acceptable implementation. To help you in attaining compliance the PCI Council supplies a wide range of documentation on its Internet web site. Here you can find the DSS standard, the intent behind the standard's requirements, and what the basic criteria are to be compliant. This is a good place to start in understanding DSS compliance and how to become/stay compliant before you begin reading about OpenEdge features and options.

http://www.pcisecuritystandards.org/

The DSS references another industry standard, OWASP[5] for application security best practices. The OWASP security documents complement the DSS standard because they address Internet-facing web applications, which are a primary target of intruders. In turn, the OWASP refers to the DSS as an example of the best practices in securing private data. For the purposes of this paper, OWASP will be treated as just another

[5] *Open Web Application Security Project*

set of PCI-DSS requirements. You can find more information about OWASP and download its documents at this URL:

http://www.owasp.org/

As with any standard, there are gray areas. There are a number of forums and books written to help you interpret the DSS in finer detail. You may also have the option of engaging a QSA to do an evaluation of your application. There are a number of other Internet sites that contain DSS-related information. Among them are sites that host information exchanges between people who have, or will, be implementing DSS compliance.

OPENEDGE AND PCI-DSS REQUIREMENTS

Using OpenEdge will not automatically make your OpenEdge application PCI-DSS-compliant. Nor will it prohibit your application from becoming compliant. OpenEdge has some features that need only be configured to meet certain DSS requirements. Other OpenEdge features will supply you with tools, or core technologies, that you can incorporate into your application and supply your own requirement solution. In some cases, OpenEdge may not supply anything that will help you meet certain requirements. The following table contains the payment card application-related PCI-DSS requirements and what options you have in using OpenEdge to meet those requirements. The table contains three columns of information: the PCI-DSS requirement number, The PCI Council-supplied description of the requirement, and information regarding how OpenEdge can assist you in meeting the requirement.

PCI-DSS REQUIREMENT #		PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
BUILD AND MAINTAIN A SECURE NETWORK			
1.	Install and maintain a firewall configuration to protect data		
	1.1–1.4	Firewalls are computer devices that control computer traffic allowed between a company's network (internal) and untrusted networks (external), as well as traffic into and out of more sensitive areas within a company's internal trusted network. The cardholder data environment is an example of a more sensitive area within the trusted network of a company. . . .	There are three instances where OpenEdge technology and firewall configurations can be at odds: > Firewalls block the NameServer's UDP protocol > Firewalls block an application server's state-reset and state-aware port ranges > Firewalls block a Progress® WebSpeed® server's port ranges Options: 1. Use direct addressing instead of using a NameServer when connecting to an application server 2. Define a small max and min port range in the application server and then open just those ports in the firewall (if you are allowed)
2.	Do not use vendor supplied defaults for system passwords and other security parameters		
	Malicious individuals (external & internal) often use vendor default passwords and other vendor defaults settings to compromise systems. These passwords and settings are well known in hacker communities and easily determined via public information.		
	2.1	Always change vendor-supplied defaults before installing a system on the network	OpenEdge uses many built-in default security parameters that should be removed or disabled: Options: 1. Disable defaults on all ABL clients, SQL server, OE Explorer/Manager, OE RDBMS and application servers. i. ABL blank userid ii. ABL security administrator iii. SQL DBA accounts iv. OEE & OEM user accounts v. AppServer process userid vi. Progress® WebSpeed® process userid vii. ABL table & field permissions viii. ABL compile time security
	2.2	Develop configuration standards for all system components. Assure these standards address all known security vulnerabilities and are consistent with industry accepted system hardening standards	
	2.2.3	Configure system security parameters to prevent misuse	
	2.2.4	Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems, and unnecessary web servers	

PCI-DSS REQUIREMENT #		PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
			ix. WebService Adapter admin userid x. OpenEdge database utility userid xi. OpenEdge database server userid 2. Set OS protection on DLC & DLC/properties directory 3. Remove OE procedure libraries that are not used by your application
PROTECT CARDHOLDER DATA			
3.	Protect stored cardholder data		
	Protection methods such as encryption, truncation, masking, and hashing are critical components of cardholder data protection. If an intruder circumvents other network security controls and gains access to encrypted data, without the proper cryptographic keys, the data is unreadable and unusable to that person. Other effective methods of protecting stored data should be considered as potential risk-mitigating opportunities. For example: *methods for minimizing risk include not storing cardholder data unless absolutely necessary, truncating cardholder data if full Primary Account Number (PAN) is not needed, and not sending PAN in unencrypted e-mails.*		
	3.3	Mask PAN when displayed	OpenEdge provides you the ability to programmatically mask the Primary Account Number information: Options: 1. Write ABL business logic to use language masking feature for user interfaces * 2. If OpenEdge SQL server is used, write SQL [Java] stored procedures to intercept the record sets and mask card data fields. * Note: ABL applications that supply this functionality need to ensure that the application's r-code is the only r-code that can connect to and access the card data. You may refer to the DBAUTHKEY for a possible solution.
	3.4	Render PAN, at minimum, unreadable anywhere it is stored (including portable digital media, backup media, in logs) by using: > One-way hash > Truncation > Index tokens/pads > Strong cryptography	OpenEdge offers more than one possible solution to encrypting card data storage: Options: 1. Use OpenEdge RDBMS TDE (Transparent Data Encryption) 2. Use a DSS-certified 3rd-party service provider 3. Write ABL application cryptography, encryption key storage*

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
				4. Write ABL application and use a 3rd-party cryptography shared library * Note: OpenEdge SQL does not have built-in cryptography services that are compatible with the ABL clients.
	3.5		Protect encryption keys used for encryption of cardholder data against both disclosure and misuse:	OpenEdge supplies a DSS-compliant key store in its data storage encryption solution, but you also have the option of providing an ABL application solution Also: Options: 1. Use OpenEdge RDBMS TDE (Transparent Data Encryption) key store 2. Write your own DSS-compliant secure key store using the ABL language* 3. Write ABL application support for integrating a secure 3rd-party key store product * Note: OpenEdge SQL does not have built-in cryptography services that would be capable of accessing a DSS compliant ABL key store.
		3.5.1	Restrict access to cryptographic keys to the fewest number of custodians necessary	
		3.5.2	Store cryptographic keys securely in the fewest possible locations and forms	
	3.6	3.6.1	Generation of strong cryptographic keys	OpenEdge supplies a DSS-compliant encryption key generation and storage in its data storage encryption solution. You also have the option of providing an ABL application solution. Also: Options: 1. Use OpenEdge RDBMS TDE (Transparent Data Encryption) encryption key generation 2. Use the ABL language's encryption key generation to support an ABL written key store* 3. Write ABL application support for integrating a secure 3rd-party encryption key generation product * Note: OpenEdge SQL does not have built-in cryptography services that would be capable of accessing a DSS-compliant ABL key store.
		3.6.2	Secure cryptographic key distribution	
		3.6.3	Strong cryptographic key storage	
		3.6.4	Periodic cryptographic key changes > As deemed necessary > At least annually	
		3.6.5	Retirement or replacement of old or suspected compromised cryptographic keys	
		3.6.6	Split knowledge and establishment of dual control of cryptographic keys	
		3.6.7	Prevent unauthorized substitution of cryptographic keys	

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
4.	Encrypt transmission of cardholder data across open, public, networks			
	Sensitive information must be encrypted during transmission over networks that are easy and common for a hacker to intercept, modify, and divert data while in transit.			
	4.1		Use strong cryptography and security protocols such as SSL/TLS to safeguard sensitive cardholder data during transmission over open, public networks	OpenEdge supports SSL/TLS on the majority of its network connections: Options: 1. Enable SSL/TLS on OpenEdge network connections * 2. Use OS IPSEC (where available) functionality 3. Use a 3rd-party SSL tunneling product * Note: Not available for Progress® OpenEdge® Replication
MAINTAIN A VULNERABILITY MANAGEMENT PROGRAM				
6.	Develop and maintain secure systems and applications			
	Unscrupulous individuals use security vulnerabilities to gain privileged access to systems. Many of these vulnerabilities are fixed by vendor-provided security patches. All systems must have the most recently released, appropriate software patches to protect against exploitation by employees, external hackers, and viruses.			
	6.1		Ensure all system components and software have the latest vendor-supplied patches installed. Install critical security patches within one month of release	OpenEdge regularly publishes service packs and maintenance releases that include all the latest security fixes. Options: 1. Use the latest version of OpenEdge
	6.3	6.3.4	Production data are not used for testing or development	OpenEdge provides data dump/load and database copy/backup tools for transporting a database from one location to another. Options: 1. Develop your own tools and processes for masking card data in a copy of the production database before it is provided to developers or technical support staff
	6.4		Follow change control procedures for all changes to system components	Progress® Developer Studio for OpenEdge® provides an extensible development framework that can interface with different types of change control systems. Options: 1. Use a secure source code control system in your development process, and require user authentication and authorization to all source code

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
	6.5	6.5.2	Injection flaws, particularly SQL injection. Also consider LDAP and Xpath injection flaws as other injection flags	The OpenEdge ABL language is not subject to the same type of injection attacks as the SQL language is. Options: 1. Write an ABL code that scans every user input source for injection flaws Note: OpenEdge SQL does not provide built-in support for preventing SQL injection attacks.
		6.5.8	Insecure cryptographic storage	OpenEdge application servers can support cryptography that can be used in securing client identity data and prevent impersonation attacks. Options: 1. Extend your OpenEdge application to encrypt the user identity information used during user logins and in authentication tokens (such as cookies) during a user session 2. Extend your OpenEdge application to use a 3rd-party product that provides secure encryption key storage 3. Use SSL/TLS to secure SQL client-server communications whenever card data or user identity information is exchanged
		6.5.9	Insecure communications	OpenEdge supports SSL/TLS on the majority of its network connections. Options: 1. Use built-in SSL/TLS network connections 2. Use a 3rd-party SSL tunneling product 3. Use OS IPSEC functionality (where available)

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
IMPLEMENT STRONG ACCESS CONTROL MEASURES				
7.	Restrict access to cardholder data by business need to know			
	To ensure critical data can only be accessed by authorized personnel, systems and processes must be in place to limit access based on need to know and according to job responsibilities.			
	7.1		Limit access to system components and cardholder data to only those individuals whose job requires such access. Access limitations must include:	Note: Compliance requires run-time access controls to database storage of cardholder data based on the user authentication in requirement #8. In OE this can be done using: 1. 100% ABL application solution 2. Split OpenEdge / ABL application solution Note: ABL application solutions must ensure that the application's r-code is the only possible r-code connection to the cardholder data. If your application does not, you may use the DBAUTHKEY password feature to provide password database access.
		7.1.1	Restriction of access rights to privileged user IDs to least privileges necessary to perform job responsibilities	OpenEdge supplies the tools that your application can use in managing user identities. Options: 1. Use the OpenEdge database Run-time Permissions option for ABL applications. Note: Requires use of the ABL CLIENT-PRINCIPAL or use of _user table accounts 2. Use the SQL Privileges for SQL clients. Note: Requires use of the _user table account 3. Write ABL application security into your application (see note in 7.1)
		7.1.2	Assignment of privileges is based on individual personnel's job classification and function	OpenEdge includes support for individual user account access controls via its Data Administration utility. Options: 1. Extend your OpenEdge application to include your own ABL role authorization functionality Note: OpenEdge SQL allows only individual user account assignments

PCI-DSS REQUIREMENT #		PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
	7.2	Establish an access control system for systems components with multiple users that restricts access based on a user's need to know and is set to "deny-all" unless specifically allowed	The OpenEdge ABL can support a "deny-all" security model based on the configuration of table/field permissions. Options: 1. Use the OpenEdge database Run-time Permissions option for ABL applications. Configure the ABL permissions for cardholder data to use the "deny-all" model. Note: Requires use of the ABL CLIENT-PRINCIPAL or use of _user table accounts. 2. Configure OpenEdge SQL server table and field privileges to cardholder data. Note: Requires the use of _user table accounts 3. Write ABL application security into your application (see note in 7.1)
8.	Assign a unique ID to each person with computer access		
	Assigning a unique identification (ID) to each person with access assures that each individual is uniquely accountable for his or her actions. When such accountability is in place, actions taken on critical data and systems are performed by, and can be tied to, known and authorized users.		
	8.1	Assign all users unique ID before allowing them to access system components or cardholder data	Note: *Requirement #8 is tightly tied to requirements #7 (authorization) and #10 (auditing).* OpenEdge includes a basic user account system that is shared between its ABL and SQL clients. That user account system is not compliant. Options: 1. Use your own ABL user accounts that support all of the requirements listed in this section. Note: Use the CLIENTPRINCIPAL to integrate with the OpenEdge database's ABL Run-time Permissions 2. Extend your ABL application to use a compliant 3^{rd}-party user account system (such as LDAP or Active Directory) Note: OpenEdge SQL will only work with _user table accounts, which will not satisfy the #8 requirements.
	8.2	In addition to assigning a unique ID, employ at least one of the following methods to authenticate all users > Password passphrase > Two-factor authentication (tokens, smart cards, etc.)	
	8.3	Incorporate two-factor authentication to remove access (network-level access from outside the network) to the network by employees, administrators, and third parties	
	8.4	Render all passwords unreadable during transmission and storage on all system components using strong cryptography	
	8.5	Ensure proper user authentication and password management for non-consumer user and administrators on all system components	

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
		8.5.1	Control addition, deletion, and modification of user IDs, credentials and other objects	
		8.5.2	Verify user identity before performing password resets	
		8.5.3	Set first-time passwords to a unique value for each user and change immediately after the first use	
		8.5.4	Immediately revoke access for any terminated users	
		8.5.5	Remove inactive user accounts at least every 90 days	
		8.5.6	Enable accounts used by vendors for remote maintenance only during the time period needed	
		8.5.8	Do not use group, shared, or generic accounts and passwords	
		8.5.9	Change user passwords at least every 90 days	
		8.5.10	Require a minimum password length of at least seven characters	
		8.5.11	Use passwords containing both numeric and alphanumeric characters	
		8.5.12	Do not allow an individual to submit a new password that is the same as any of the last four passwords he or she has used	
		8.5.13	Limit repeated access attempts by locking out the user ID after not more than six attempts	
		8.5.14	Set the lockout duration to 30 minutes or until the administrator enables the user ID	
		8.5.15	If a session has been idle for more than 15 minutes, require the user to re-enter the password to re-activate the terminal	
		8.5.16	Authenticate all access to any database containing cardholder data. This includes access by applications, administrators, and other users	

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
REGULARLY MONITOR AND TEST NETWORKS				
10.	Track and monitor all access to network resources and cardholder data			
	Logging mechanisms and the ability to track user activities are critical. The presence of logs in all environments allows thorough tracking and analysis if something does go wrong. Determining the cause of a compromise is very difficult without system activity logs.			
	10.1		Establish a process for linking all access to system components (especially access done with administrative privileges such as root) to each individual user	Note: *Requirement 10 is tightly tied to the user authentication in requirement #8.* OpenEdge supplies a secure, nonreputable, auditing feature for tracking privileged and non-privileged user operations in the ABL & SQL languages and the RDBMS utilities. Options: 1. Write an ABL job that archives the captured OpenEdge auditing data to an external audit product that is common to the enterprise
	10.2		Implement automated audit trails for all system components to reconstruct the following events:	OpenEdge's auditing feature supplies secure auditing of all requirement line items. Options: 1. Configure and use OpenEdge auditing support for database CUD operations for ABL & SQL languages 2. Write ABL database schema triggers for ABL application R operations (to augment #1) 3. Use the OpenEdge supplied auditing policies for administrator and database utility auditing Note: OpenEdge SQL does not have the ability to trap and record SELECT operations.
		10.2.1	Access to cardholder data	
		10.2.2	Actions taken by administrators with root or administrative privileges	
		10.2.3	Access to audit trails	
		10.2.4	Invalid logical access attempts	
		10.2.5	Use of identification and authorization mechanisms	
		10.2.7	Creation and deletion of system-level objects	
	10.3		Record at least the following audit trail entries for all system components for each event:	The OpenEdge auditing feature records information that identifies the user, the date the event occurred, and the affected object. Options: 1. Configure and use OpenEdge auditing support
		10.3.1	User identification	
		10.3.2	Type of event	
		10.3.3	Date and time	
		10.3.4	Success or failure indication	
		10.3.5	Origination of event	
		10.3.6	Identity of name of affected data, system component or resource	

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
	10.5		Secure audit trails so they cannot be altered	OpenEdge's auditing feature incorporates strict access controls on audit administration, access, and maintenance. Separation of Duty is also available. The ability to update audit data is prohibited and the ability to delete audit data is severely restricted. Options: 1. Configure and use OpenEdge auditing support
		10.5.1	Limit viewing of audit trails to those with a job-related need	
		10.5.2	Protect audit trail files from unauthorized modifications	
		10.5.3	Promptly backup audit trail files to a centralized log server or media that is difficult to alter	
		10.5.4	Copy logs for wireless networks onto a log server on the internal LAN	
		10.5.5	Use file integrity monitoring and changed detection software on logs to ensure that existing log data cannot be changed without generating alerts	
	10.6		Review logs for all system components at least daily. Log reviews must include those servers that perform security functions like IDS, authentication, authorization, and accounting protocol	OpenEdge auditing data may be accessed by an authorized user from either ABL or SQL clients. Options: 1. Run OpenEdge supplied reports for internal OpenEdge audited events 2. Write customized ABL reports for examining audit trail data 3. Use OpenEdge SQL server and a 3rd party report generator to query and report audit trail data
	10.7		Retain audit trail history for at least one year, with a minimum of 3 months on-line availability.	OpenEdge audit data lifecycle is under the sole control of the audit administration accounts. Options: 1. Create and use an OpenEdge database with large capacity for long term (1 year) storage. Use OpenEdge archive tool to frequently archive audit trail data from the production database to the long-term audit storage database. Use the OpenEdge archive tool to archive long-term database storage to off-site storage 2. Write customized ABL archive utility to filter and manage short-term and long-term storage locations

PCI-DSS REQUIREMENT #			PCI-DSS VERSION 1.2 REQUIREMENT DESCRIPTION	OPENEDGE SUPPORT COMMENTS AND OPTIONS
APPENDIX A: PCI DSS APPLICABILITY FOR SHARED HOSTING PROVIDERS				
A.	Hosting providers protect cardholder data environment			
	As referenced in requirement 12.8, all service providers with access to cardholder data (including shared hosting providers) must adhere to the PCI-DSS. In addition, requirement 2.4 states that shared hosting providers must protect each entity's hosted environment and data. Therefore, shared hosting providers must additionally comply with the requirements in this appendix.			
	A.1		Protect each entity's (merchant, service provider, or other entity) hosted environment and data. A hosting provider must fulfill requirements A.1 thru A.4 as well as all other relevant sections of the PCI-DSS Note: *Even though a hosting provider may meet these requirements, the compliance of the entry that uses the hosting provider is not guaranteed. Each entity must comply with the PCI-DSS and validate compliance as applicable.*	OpenEdge supports many different architectures that can be applied to a service provider model. Options: 1. Use a separate OpenEdge application installation per customer. A typical installation would include: i. AppServer ii. OpenEdge database iii. OEM/OEE iv. Application server clients v. OpenEdge application utilities
		A.1.1	Ensure each entity only runs processes that have access to that entity's cardholder data environment	
		A.1.2	Restrict each entity's access and privileges to own cardholder data environment	
		A.1.3	Ensure logging and audit trails are enabled and unique to each entity's cardholder data environment and consistent with PCI-DSS requirement 10	
		A.1.4	Enable processes to provide for timely forensic investigation in the event of a compromise to any hosted merchant or service provider	

SUMMARY

Having a PCI-DSS-compliant OpenEdge application is an advantage in the business application market. Not only does it mean your OpenEdge application can handle credit card transactions; it means that your application has met the criteria for a secure application in any production environment. Once you have made the initial transition to PCI-DSS compliance, it will be incremental work from that point forward. What it means to make that initial transition is dependent on your application's architecture and implementation.

OpenEdge lessens the impact of becoming PCI-DSS-compliant by providing a rich environment in which you can craft your implementation of PCI-DSS requirements within the bounds of your application's architecture. In addition, OpenEdge continues to provide new solutions, such as OpenEdge auditing and Transparent Database Encryption, to meet some of the harder requirements for you. OpenEdge continues to be the platform that helps you meet all types of business application needs.

PROGRESS SOFTWARE

Progress Software Corporation (NASDAQ: PRGS) is a global software company that enables enterprises to be operationally responsive to changing conditions and customer interactions as they occur. Our goal is to enable our customers to capitalize on new opportunities, drive greater efficiencies, and reduce risk. Progress offers a comprehensive portfolio of best-in-class infrastructure software spanning event-driven visibility and real-time response, open integration, data access and integration, and application development and management—all supporting on-premises and SaaS/cloud deployments. Progress maximizes the benefits of operational responsiveness while minimizing IT complexity and total cost of ownership.

WORLDWIDE HEADQUARTERS

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA
Tel: +1 781 280-4000 Fax: +1 781 280-4095 On the Web at: www.progress.com

Find us on facebook.com/progresssw twitter.com/progresssw youtube.com/progresssw

For regional international office locations and contact information, please refer to the Web page below:
www.progress.com/worldwide

PROGRESS software
BUSINESS MAKING PROGRESS™