Australian Government

Department of Agriculture,
Fisheries and Forestry

NEXDOC

VIP03 - Vendor Software Requirements

Document Properties

Document version	1.1
Document status	**Final**
Issue date	June 2023
Print date	June 2023
Author	Department of Agriculture, Fisheries and Forestry

Glossary

Term	Meaning
Client Group	The entity that lodges and manages REX. The initial owner of a REX is the Client Group that lodged it. A REX can only be owned by one Client Group at any one point in time. A Client Group can have one or more individuals as its members.
Client Group token	A 32-character string of numbers and letters that uniquely identifies a Client Group. All messages that are sent NEXDOC web services that relate to REXs **must** contain a Client Group token to identify the Client Group who is performing the operation.
Client Group ID	An identifier that uniquely identifies a Client Group. The owner of a Client Group can share its Client Group ID with other entities for transferring and forwarding REXs and for printing Certificates.
Client Group member	A person who is a member of one or more Client Groups. This person uses third party software or the NEXDOC web portal to lodge and manage REXs. To use NEXDOC web portal to lodge and manage REXs, the person needs to have registered a user account to use the Department's online services and needs to have registered as an exporter.
Client token (aka Client Group member token)	A 32-character string of numbers and letters that uniquely identifies a person who can be a member of one or more Client Groups. All messages that are sent NEXDOC web services that relate to REXs **must** contain a Client token to identify the specific person who is performing the operation. The person identified by the Client token in the web service request, **must** be a member of the Client Group identified by its Client Group token in the web service request.
Client Group administrator	A person who has been given permission to create and manage Client Groups and Client Group members of their organisation.
Installation token (aka web service user)	A service account that should be included in a message send to Department's SOAP and REST web services for authentication. Typically, there should be one service account per installation of third party software that consumes the Department's web services. As such the web service user ID and password pair is also called an Installation token.
Installation administrator	A person who can create and managed web service users. Web service users created by an Installation administrator can only be managed by Departmental staff and by the same Installation administrator or other Installation administrators in the same organisation.
Exporter	The entity that owns the consignment for which the export documentation is being requested. An Exporter can be an individual or a business.
Exporter ID	An identifier that uniquely identifies an Exporter. The Exporter ID is automatically generated when an Exporter record is created. The individual who owns an Exporter can share its Exporter ID with their agents.
Third party software vendor	A software developer (individual or business) who develops software to integrate with NEXDOC web services.

Contents

1 Overview

1.1 Purpose

This document describes the NEXDOC and its associated system requirements that vendors must comply with, including in relation to disclaimers and compliance, terms and conditions, and acceptable use of Department's resources.

1.2 Document scope

This document stipulates vendor requirements for utilising NEXDOC services.

1.3 Capabilities

The requirements fall into four capability categories:

1) General Requirements

2) Compliance Declarations and Disclaimers

3) Declarations Codeset

4) Security and Audit Controls

2 General Requirements

GVR.01	All operations that are performed by the vendor's software **should** be logged for 12 years for auditing purposes with an easy to access interface.
GVR.02	Vendors **should** develop their software based on the latest software development guide as published by the Department.
GVR.03	Vendor Software **must** successfully complete the test cases provided in the most recent version of the conformance suite. Vendors will have 6 weeks to become compliant with new changes issued by the Department, unless otherwise specified by the Department. Note: Under the Software Vendor Terms and Conditions, failure of Software Vendors to become compliant with this requirement may result in the Department denying Software Vendors access to the NEXDOC production environment.

3 Compliance Declarations and Disclaimers

Identifier	Description
CDD.01	Software Vendors **should** provide declarations, disclaimers and notices in their software to make users aware of their obligations as stated in the software development guide and other reference material provided by the Department and seek acceptance to declarations and notices where required. Declarations, disclaimers and notices **must** be in the form required by the Department. Software Vendors **must** make users aware of declarations, disclaimers and notices, including by clearly displaying the relevant declaration, ***disclaimer*** or notice to users before they are able to perform the function to which the relevant declaration, disclaimer or notice relates (eg lodge a REX, remote print).
CDD.O2	The following ***disclaimer*** for remote print **must** be shown and agreed to by all users when attempting to remote print a document provided by NEXDOC. To be able to print export documents at my own site, I agree to the following: • The print site is located in Australia • I have a software package to communicate with NEXDOC that has been registered by the Department • I have a duplex laser printer which supports at least 600 dpi • I will only print one set of export documents for each consignment • I will not alter, vary or in any way change an export document generated in NEXDOC once it has been printed • I understand that there are a number of criminal offences under the Criminal Code Act 1995 (Cth) relating to the making and use of a false Commonwealth document. • Inability to demonstrate valid and legitimate use of the printed export document(s) may result in the Department revoking the option to print NEXDOC generated document(s) through the remote print function.
CDD.03	The Software **must** display the following text for **‘D’ (Dairy), ‘E’ (Eggs) or ‘F’ (Fish)** commodities when collecting the data for this segment: Is the Exporter in possession of either: • a declaration that complies with clause 6 of Schedule 9 of the relevant Orders; or • a written verification by an authorised officer made under clause of Schedule 9 of the relevant Order? [**Y**(Yes) or **N**(No)]

CDD.04	The Vendor Software **must** display the following text for **‘G’ (Grains)** (Horticulture) commodities when collecting the data for this segment: Do you declare that – • the conditions or restrictions prescribed in Regulations or Orders under the Export Control Laws and applicable to the goods been complied with; and • the information supplied on this form is true and correct in every particular. Note: For criminal penalties applying to persons who make false misleading statements to a Commonwealth entity see the Criminal Code Act 1995 Part 7.4 (false or misleading statements).

Identifier	Description
CDD.05	The Vendor Software **must** display the following text for **‘M’ (Meat)** commodity when collecting the data for this segment: • Do you have effective measures in place to ensure there is a sound basis for the information provided in this permit application? Note: For criminal penalties applying to persons who make false misleading statements to a Commonwealth entity see the Criminal Code Act 1995, Part 7.4 (false or misleading statements).
CDD.06	The Vendor Software **must** display of the following message prior to transmission of data to NEXDOC: **WARNING.** You are giving information to a Commonwealth entity. Giving false or misleading information to a Commonwealth entity is a serious offence. There is a risk that information sent to or from the departmental email account (NEXDOC@agriculture.gov.au) may be viewed, intercepted, corrupted or modified by, or otherwise disclosed to, third parties. The Commonwealth cannot guarantee or otherwise warrant the security or integrity of any information transmitted in this manner and is not liable in connection with the transmission of any such information (including as a result of negligence).
CDD.07	The Vendor Software **must** include a: Declaration of Compliance Indicator **(a1)** • **Y** (Yes) means the Exporter has declared that they are in a possession of a Declaration of Compliance for the products being exported **N** (No) the converse
CDD.08	The Vendor Software **must** include an: Imported Flag Indicator **(a1 - Dairy)** • **Y** (Yes) indicates that the Exporter has declared that products being exported contain imported dairy ingredients, other than from New Zealand • **N** (No) the converse

CDD.09	The Vendor Software **must** include an: Imported Flag Indicator **(a1 - Fish)** • **Y** (Yes) indicates that the Exporter has declared that the REX includes imported product • **N** (No) the converse
CDD.10	The Vendor Software **must** include a: True and complete indicator **(a1)** • **Y** (Yes) indicates that the Exporter has declared that the information provided in the application for an export permit is true and complete • **N**(No) the converse

4 Declarations Code sets

Declarations Code sets are now provided in separate files in recognition that they are subject to more frequent changes than the other requirements defined in this document. These code sets are now provided within the following CSV file:

NEXDOC VIP03 - Certificate Declarations Code Set v1.0

This will be distributed alongside this document.

Software vendors are required to display the most up-to-date declarations information as provided by the department.

5 Security Compliance

Vendors must provide a Statement of Compliance regarding the following security requirements. Where a Vendor is not compliant with all mandatory requirements (indicated by the word **must**) the Vendor must also supply a development roadmap indicating the timeframes for achieving compliance. The Statement of Compliance (and compliance roadmap where required) will be assessed for overall risk by the Department. Vendors that do not meet the Department's standards of compliance will not be granted access to the NEXDOC Production environment.

The security requirements below are required in order to address the following identified risks:

- External malicious threat actor's deliberately bypassing web portal and web services security controls, gaining code execution on 3rd party software provider servers.
- External malicious threat actor's deliberately bypassing web portal and web services security controls, gaining code execution and successfully escalating privileges on 3rd party Software Provider servers and pivoting onto 3rd Software Provider enterprise network.
- Unauthorised 3rd party Software Provider NEXDOC proxy system access and information disclosure by external users.
- Unauthorised 3rd Party Software Provider and NEXDOC proxy system access and information disclosure by internal users.

Where alternate or additional methods have been used to address these risks the Vendor should include this information in the Statement of Compliance.

ASVS ID	Requirements	Compliant Y/N
5.1	**Input Validation Requirements**	
5.1.1	The application **must** have defences against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, or environment variables)	
5.1.2	The framework **must** protect against mass parameter assignment attacks, and have countermeasures to protect against unsafe parameter assignment, such as marking fields private or similar.	
5.1.3	All input (HTML form fields, REST requests, URL parameters, HTTP headers, cookies, batch files, RSS feeds, etc) **should** validate using positive validation (whitelisting)	
5.1.4	Structured data **must** be strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers or telephone, or validating that two related fields are reasonable, such as checking that suburb and zip/postcode match)	
5.1.5	URL redirects and forwards **must** only allow whitelisted destinations or show a warning when redirecting to potentially untrusted content.	
5.2	**Sanitization and Sandboxing Requirements**	
5.2.1	All untrusted HTML input from WYSIWYG editors or similar **must** be properly sanitized with an HTML sanitizer library or framework feature	
5.2.2	Unstructured data **must** be sanitized to enforce safety measures such as allowed characters and length	
5.2.3	The application **must** sanitize user input before passing onto mail systems to protect against SMTP or IMAP injection.	

ASVS ID	Requirements	Compliant Y/N
5.2.4	The application **must** avoid the use of eval() or other dynamic code execution features (Where there is no alternative, user input **must** be included to be either being sanitized or sandboxed before being executed	
5.2.5	The application **should** protect against template injection attacks by ensuring that any user input being included is sanitized or sandboxed.	
5.2.6	The application **must** protect against SSRF attacks, by validating or sanitizing untrusted data or HTTP file metadata, such as filenames and URL input fields, use whitelisting of protocols, domains, paths and ports.	
5.2.7	The application **should** sanitize, disable, or sandbox any user-supplied SVG scriptable content, especially as they relate to XSS resulting from inline scripts, and foreignObject.	

5.2.8	The application **should** sanitize, disable, or sandbox any user-supplied scriptable or expression template language content, such as Markdown, CSS or XSL stylesheets, BBCode, or similar.	
5.3	**Output encoding and Injection Prevention Requirements**	
5.3.1	Output encoding **should** be relevant for the interpreter and context required? (For example, use encoders specifically for HTML values, HTML attributes, JavaScript, URL Parameters, HTTP headers, SMTP, and others as the context requires, especially from untrusted inputs (e.g. names with Unicode or apostrophes, such as ねこ or O'Hara).)	
5.3.2	Output encoding **should** preserve the user's chosen character set and locale, such that any Unicode character point is valid and safely handled.	
5.3.3	Context-aware output escaping (preferably automated - or at worst, manual) **must** protect against reflected, stored, and DOM based XSS.	
5.3.4	Data selection or database queries (e.g. SQL, HQL, ORM, NoSQL) **must** use parameterized queries, ORMs, entity frameworks or other means, to protect from database injection attacks.	
5.3.5	Where parameterized or safer mechanisms are not present, Context-specific output encoding **must** be used to protect against injection attacks, such as the use of SQL escaping to protect against SQL injection.	
5.3.6	The application **must** provide protection against JavaScript or JSON injection attacks, including for eval attacks, remote JavaScript includes, CSP bypasses, DOM XSS, and JavaScript expression evaluation.	
5.3.7	The application **should** provide protection against LDAP Injection vulnerabilities, or have specific security controls in place to prevent LDAP Injection.	
5.3.8	The application **should** provide protection against OS command injection. And, for OS System calls, parameterized OS queries or use contextual command line output encoding **should** be used.	
5.3.9	The application **must** protect against Local File Inclusion (LFI) or Remote File Inclusion (RFI) attacks.	
5.3.10	The application **should** provide protection against XPath injection or XML injection attacks.	
5.5	**Deserialization Prevention Requirements**	

ASVS ID	Requirements	Compliant Y/N
5.5.1	Serialized objects **must** use integrity checks or encryption to prevent hostile object creation or data tampering.	
5.5.2	The application **must** correctly restrict XML parsers to only use the most restrictive configuration possible, and ensure that unsafe features, such as resolving external entities, are disabled to prevent XXE.	

5.5.3	Deserialization of untrusted data **must** be avoided or **must** be protected in both custom code and third-party libraries (such as JSON, XML and YAML parsers)	
5.5.4	When parsing JSON in browsers or JavaScript-based backends, JSON.parse **should** be used to parse the JSON document. However, do not use eval() to parse JSON.	
13.3	**SOAP Web Service Verification Requirements**	
13.3.1	XSD schema validation **should** take place to ensure a properly formed XML document, followed by validation of each input field before any processing of that data takes place.	
2.1	**Password Security Requirements**	
2.1.1	User set passwords **should** be at least 12 characters in length.	
2.1.2	Passwords that are 64 characters or longer **should** be permitted.	
2.1.3	Passwords **should** not contain spaces and truncation.	
2.1.4	Unicode characters **should** be permitted in passwords. A single Unicode code point is considered a character, so 12 emoji or 64 kanji characters **should** be valid and permitted.	
2.1.5	Users **must** be able to change their password.	
2.1.6	Password change functionality **must** require the user's current and new password.	
2.1.7	Passwords submitted during account registration, login, and password change **should** be checked against a set of breached passwords either locally (such as the top 1,000 or 10,000 most common passwords which match the system's password policy) or using an external API. If the password is breached, the application **must** require the user to set a new non-breached password.	
2.1.8	The application **should** have a password strength meter provided to help users set a stronger password.	
2.1.12	The user **should** be able to choose to either temporarily view the entire masked password, or temporarily view the last typed character of the password on platforms that do not have this as native functionality.	
2.2	**General Authenticator Requirements**	
2.2.1	Anti-automation controls **must** be effective at mitigating breached credential testing, brute force, and account lockout attacks. Such controls include blocking the most common breached passwords, soft lockouts, rate limiting, CAPTCHA, ever increasing delays between attempts, IP address restrictions, or risk-based restrictions such as location, first login on a device, recent attempts to unlock the account, or similar. Verify that no more than 100 failed attempts per hour is possible on a single account.	
2.2.3	Secure notifications **must** be sent to users after updates to authentication details, such as credential resets, email or address changes, logging in from	

ASVS ID	Requirements	Compliant Y/N
	unknown or risky locations. The use of push notifications - rather than SMS or email - is preferred, but in the absence of push notifications, SMS or email is acceptable as long as no sensitive information is disclosed in the notification.	
2.3	**Authenticator Lifecycle Requirements**	
2.3.1	System generated initial passwords or activation codes **should** be securely randomly generated, **must** be at least 6 characters long, and **must** contain letters and numbers, and expire after a short period of time. These initial secrets **must** not be permitted to become the long-term password.	
2.5	**Credential Recovery Requirements**	
2.5.1	A system generated initial activation or recovery secret **should** not be sent in clear text to the user.	
2.5.2	Password hints or knowledge-based authentication (so-called "secret questions") **should** not be present.	
2.5.3	Password credential recovery **must** not reveal the current password in any way.	
2.5.4	Shared or default accounts **must** not be present (e.g. "root", "admin", or "sa").	
2.5.5	If any authentication factor is changed or replaced, the user **must** be notified of this event.	
2.5.6	Forgotten passwords, and other recovery paths **must** use a secure recovery mechanism, such as TOTP or other soft token, mobile push, or another offline recovery mechanism.	
2.7	**Out of Band Verifier Requirements**	
2.7.2	The out of band verifier **must** expire out of band authentication requests, codes, or tokens after 10 minutes.	
2.7.3	Out of band verifier authentication requests, codes, or tokens **must** only be usable once, and only for the original authentication request.	
4.1	**General Access Control Design**	
4.1.1	The application **must** enforce access control rules on a trusted service layer, especially if client-side access control is present and could be bypassed.	
4.1.2	All user and data attributes and policy information used by access controls **must** not be able to be manipulated by end users unless specifically authorized.	
4.1.3	The principle of least privilege **must** exist within the application - users **should** only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege.	
4.1.4	The principle of deny by default **must** exist whereby new users/roles start with minimal or no permissions and users/roles **must** not receive access to new features until access is explicitly assigned.	
4.1.5	Access controls **must** fail securely including when an exception occurs.	

4.2	**Operation Level Access Control**	
4.2.1	Sensitive data and APIs **must** be protected against direct object attacks targeting creation, reading, updating and deletion of records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records.	
ASVS ID	**Requirements**	**Compliant Y/N**
4.2.2	The application or framework **should** enforce a strong anti-CSRF mechanism to protect authenticated functionality, and effective anti-automation or anti-CSRF **should** protect unauthenticated functionality.	