Y/IT-337

B&W Y-12 technical services LLC

Y-12 NATIONAL SECURITY COMPLEX

PRIDE Surveillance Projects Data Packaging Project

Information Package Specification Version 1.1

August 31, 2010

Matthew Kelleher

Rick Shipp
Information Technology
Y-12 National Security Complex

James David Mason
SAIC

MANAGED BY
B&W Y-12, LLC
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

UCN-13672 (1-08)

Y/IT-337

PRIDE Surveillance Projects Data Packaging Project Information Package Specification Version 1.1

August 31, 2010

Matthew Kelleher

Rick Shipp
Information Technology
Y-12 National Security Complex

James David Mason
SAIC

ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions of Rob Wilson, Richard Secrist, Monica Love, and Martin McNeil of the PRIDE Y-12 Product Characterization System Migration Project to the information package specification. PCS Migration Project team plans to move Y-12 product information from Product Characterization System to a new platform and they plan to use information packages to store that moved information. They proposed several changes to the specification that improved their ability to store moved information.

This report was prepared in XML, using the "Y-12–Report" application of Arbortext developed under the *PRIDE Surveillance Reports Collaborative Authoring Project*.

ABSTRACT

Information Package Specification version 1.1 describes an XML document format called an information package that can be used to store information in information management systems and other information archives. An information package consists of package information, the context required to understand and use that information, package metadata that describes the information, and XML signatures that protect the information. The information package described in this specification was designed to store Department of Energy (DOE) and National Nuclear Security Administration (NNSA) information and includes the metadata required for that information: a unique package identifier, information marking that conforms to DOE and NNSA requirements, and access control metadata. It is an implementation of the Open Archival Information System (OAIS) Reference Model archival information package tailored to meet NNSA information storage requirements and designed to be used in the computing environments at the Y-12 National Security Complex and at other NNSA sites.

TABLE OF CONTENTS

1. INTRODUCTION

Information Package Specification version 1.1 describes an XML document format called an information package that can be used to store information in information management systems and other information archives. An information package consists of package information, the context required to understand and use that information, package metadata that describes the information, and XML signatures that protect the information. The information package described in this specification was designed to store Department of Energy (DOE) and National Nuclear Security Administration (NNSA) information and includes the metadata required for that information: a unique package identifier, information marking that conforms to DOE and NNSA requirements, and access control metadata. Information packages also include search terms that help users locate information and documentation metadata that describes the information. Information stored in the information package can be text content, binary content, or the contents of files or other containers. All content is stored as text or in a standard text encoding of binary content such as base 64. A single information package can contain multiple types of information.

Information packages were developed by the Y-12 Data Packaging project funded by the NNSA PRIDE (Product Realization Integrated Digital Environment) program. PRIDE was created to improve the quality of digital information in the Nuclear Security Enterprise and to make that information available to all NSE users that need it. In FY 2008 PRIDE funded a project to investigate a taxonomy-based search and retrieval system to serve as a single source of Y-12 information for Y-12 and the NSE. All Y-12 information available to system users would have standardized metadata whose terms and definitions were consistent across the NSE. A user would locate Y-12 information by searching this metadata for appropriate metadata terms then following links to the information.

The project team documented the results of its investigation in Y-12 report Y/IT-193 *Taxonomy-Based Search and Retrieval Implementation at Y-12* [Y/IT-193]. The investigation showed that taxonomy-based search and retrieval was not practical at Y-12 for a number of reasons. One reason was that Y-12 information was not ready to be released to users through a taxonomy-based search and retrieval system. Since Y-12 information must eventually be made available to users through a system such as a taxonomy-based search and retrieval system, the project team recommended that:

- Information to be released to search users must be identified.
- Some information must be packaged with context information.
- Information must be assigned search metadata using terms from the standard taxonomic scheme.
- Information must have a classification/sensitivity review and appropriate markings applied.
- Information must be assigned standard access control metadata.
- Information should be stored in packages such as XML documents.

For FY 2009 the Y-12 project team requested and received funding from PRIDE for the Y-12 PRIDE Surveillance Projects Data Packaging Project . The project goal was to develop a set of prototype data packages using XML that can be used to store a representative set of Y-12 inspection data along with the

context required to understand and use the data. The team planned to show how these inspection data packages could be stored in an XML database and how XML technologies could be used to locate and extract inspection data from these inspection data packages. Extracted data would be in a form ready for use in surveillance and process improvement activities.

The project team realized that data packages can be used to store all types of information, not just inspection data, and that all information packages should have the same metadata structure. The project team developed a standard XML structure for an information package that includes all metadata required to identify, protect, locate, and describe the package information and documented this structure in report Y/IT-278 *Information Package Specification Version 1.0* [Y/IT-278]. This report contains a summary of the findings in Y/IT-193 *Taxonomy-Based Search and Retrieval Implementation at Y-12*, a set of requirements for information packages, a specification for an information package that can be translated into an XML document definition, and the background information required to understand and use information packages. The report includes a working example that shows how information packages can be stored in an XML database and the information in them retrieved and used in surveillance and process improvement activities.

The project was funded for one year and the report was issued at the end of the year. While the report was being written, the Y-12 Product Characterization System (PCS) Migration Project team used the first drafts of the specification to develop information packages to store information they plan to move from PCS to a Windchill PDMLink system. The Electronic Build History project planned to start implementing information packages in early FY 2010. The Data Packaging project team recognized that issues raised by these implementations could not be addressed in the first specification version and that a revised specification would be needed.

Information Package Specification version 1.1 described in this report resolves issues raised by these two implementations. It updates the requirements and specification, provides information on how to implement and use the specification, and includes an XML Schema definition for information packages. It does not provide information package background or the working example provided in the version 1.0 report. This information can be obtained from the version 1.0 report if needed. The working example was upgraded to version 1.1 of this specification and is available on request.

The information package described in this specification is an implementation of information packages described in the Open Archival Information System (OAIS) Reference Model [OAIS1, OAIS2]. This reference model describes a possible architecture of a complete archival information system designed to receive information to be archived, manage that information in an archive, and provide that information to users. Information is sent to the archive, stored in the archive, and provided by the archive in information packages. This information package is an implementation of the OAIS archival information package tailored to meet NNSA information storage requirements and to be used in the computing environment at Y-12 and other NSE sites. See appendix D for more information on how the information package described in this specification conforms to the OAIS Reference Model.

2. INFORMATION PACKAGE REQUIREMENTS

This section lists the requirements the information package design must meet to provide for information package identification, meet DOE requirements for information marking, provide appropriate information access control, enable users to locate information, protect information, and make information packages a practical format for storing DOE and NNSA information. Each requirement includes the justification for the requirement and describes how the requirement is met in this specification. Following sections provide more information on how these requirements are met by the specification.

1. **Information packages must store package information using an appropriate XML structure for that information.**

   An information package must be capable of storing a wide variety of information types. Each information type should be stored in an appropriate XML structure for that information type. This specification cannot anticipate the types of information that may be stored, so it cannot define standard information type XML structures. It must allow each type of information to be stored in an appropriate XML structure for the information type.

   This specification places all package information in a single child element of the document root element. This child element contains the parent elements of the package information XML structures and can contain the parent elements of the XML Signatures that protect the information. This specification does not specify the structure or contents of package information XML structures.

2. **Information packages must have common package metadata and a common XML structure for that metadata.**

   Storing common package information metadata in a common XML structure allows applications and users to locate and retrieve package metadata using the same procedures for all information packages. For example, when information packages containing different types of information are stored in a single information management system, a common package metadata structure allows the information management system to extract from these information packages the information required to manage and protect them and the information required to locate them.

   This specification places package metadata in the information package document root element and in all elements that are child elements of the information package document root element except for the element that is the parent element of the package information XML structure.

3. **Package metadata must include a unique information package identifier.**

   Every information package must have a unique identifier that serves as an unambiguous reference to the information package. This unique identifier is required so that calculations performed using package information or actions based on information in package information

can be traced back to the information packages used. The identifier also serves as a unique identifier for the information package when it is stored in an information management system.

This requirement is met in this specification by providing for a unique package identifier in the package metadata.

4. **The information package identifier must include an optional version**.

Versions may be needed in situations where package information is managed using a configuration control process that uses a version to identify the information. For example, product inspection information may be stored in an information package as soon as it is collected. That information may subsequently be added to, corrected, approved, and have a specification exception recorded for it. The configuration control process used for this information may require each version of the information package be identified and saved. To ensure that each information package can be tracked using its unique identifier, the package identifier must consist of an identifier and an optional version. The information package identifier created from this identifier and version must be unique. The version must be optional for those information packages that are not identified by a version.

This requirement is met in this specification by including an optional version in the package identifier in the package metadata.

5. **Package metadata must include information markings that conform to DOE requirements**

All Department of Energy classified and unclassified controlled information must be protected by marking it as required by DOE and other Federal agency regulations. These markings warn those in possession of classified or unclassified controlled information to release that information only to persons allowed access to and with a need-to-know that information and to store that information only on systems approved to store that information.

This requirement is met by including or providing for all required markings for classified and unclassified controlled information in the package metadata.

6. **Package metadata must include access control metadata that can be used to determine whether a user is allowed access to the package information.**

An information system may be used to store information packages that contain different types of information. Users of this information system may have a need-to-know for information in some but not all of the stored information packages. The information package design must allow the creator of each information package to specify the access control metadata the information management system needs to determine whether a user has access to package information.

This requirement is met by providing for access control information attributes in the package metadata.

7. **Package metadata must include search terms that can be used to locate package information stored in information packages.**

Information management systems used to store information packages will likely contain many types of information packages with each information package type stored using its appropriate XML structure. Without search terms, a user searching for information would have to search these structures for terms that identify the information the user needs. A standard location and format for search terms would assist the user by providing a standard location for terms that describe each information package. The user could search this standard location for some or all of the terms that identify the information needed by the user.

This requirement is met by providing for search terms in the package metadata.

8. **Package metadata must include optional references that can be used to identify related information packages.**

   Package information stored in an information package is often related to package information stored in other information packages. Providing references to related information packages would assist users in identifying and locating related package information in these information packages.

   This requirement is met by providing references to related information packages in the package metadata.

9. **Package metadata should include optional package history and optional package notes.**

   Package history and notes provide additional information that may help a user understand the information package and package information.

   This requirement is met by providing for history and notes in the package metadata.

10. **Information in information packages should be stored using formats that conform to recognized standards issued by appropriate standards institutions.**

    Information packages may be stored for decades so the information in them should be stored using formats that will remain understandable for decades. Only formats that conform to recognized standards can be expected to be understandable for this length of time. Formats used by vendor software are likely to remain understandable only as long as the vendor supports the format. Once vendor support ends, the information may quickly become unreadable. A site-specific format will remain understandable only as long as the site supports it. Once site support is lost, the information may become unreadable.

    This requirement is met by requiring that information packages be standard XML documents. This specification also recommends standard forms for dates, times, and measurement units and a standard for encoding binary data and other information not in a form compatible with XML.

11. **Information in information packages should be protected so that changes to the information can be detected.**

    Information packages will be stored on digital media that is not 100% reliable. Over time information packages stored on this media may become damaged. This damage may be hard

to detect and may significantly alter the information in the package. Some mechanism for detecting this damage should be used so users can identify and handle damaged information packages appropriately.

This requirement is met by using an XML Signature, a standard implementation of digital signatures for XML documents, to sign package information.

12. **Information in information packages should be protected so that the source of the information can be reliably identified.**

    This provides an extra level of protection from a possible mixing of original information packages and modified information packages such as test packages.

    This requirement is met by using an XML Signature to sign package information. The certificate that contains the public key used to decrypt the signature identifies the organization that signed the information.

13. **Information packages should be structured to protect package information yet allow the package metadata to be changed as needed.**

    Package information must be protected in such a way that changes to the package information can be detected and the source of package information identified. However, package metadata such as package identifier, access control, information marking, search terms, package history, and notes should be allowed to change. The package identifier can be changed to identify successor versions. Access control and information markings can change to reflect new access control and information marking requirements. New search terms may be identified and entries added to the package history.

    This requirement is met by signing only package information. The rest of the information is not protected and can be changed at any time.

3. INFORMATION PACKAGE XML DOCUMENT STRUCTURE

The information package XML document structure consists of a document root `<InfoPackage>` element, seven child elements that contain package metadata, and a single `<PackageInfo>` element that is the parent element of the package information XML document structures. This structure separates the package metadata from the package information.

The XML specification requires all XML documents be structured as a tree with a single XML element called the document root element at the base of the tree [XML 2008]. Element `<InfoPackage>` is the document root element for every information package XML document. All `<InfoPackage>` child elements except the `<PackageInfo>` element described below contain package metadata. These elements are listed below and are described in the following sections:

- `<PackageIdentification>` – Package identification and description

- <InformationMarking> – Information marking
- <AccessControl> – Access control metadata
- <SearchTerms> – Search terms
- <References> – References to other information packages
- <History> – History of the information package or package information
- <Notes> – Notes about the information package or package information

Element <PackageInfo> contains the parent elements of the package information structure and can contain an XML Signature <Signature> element used to digitally sign package information structure elements that must be protected. This specification does not specify the names or structure of the package information elements. However, it recommends that each <PackageInfo> child element have a version attribute that identifies the package information structure version. This version will help processing applications identify and process the package information structure.

XML Signatures can be used to protect some or all of the contents of <PackageInfo> and its child elements. Each XML signature is represented by a <Signature> element placed directly below the <PackageInfo> element or below a <PackageInfo> child element. A single <Signature> element can protect the <PackageInfo> element and/or some or all of its child elements. Child elements may be protected by separate <Signature> elements.

4. PACKAGE IDENTIFIERS AND DESCRIPTION

Every information package must have a unique identifier that serves as an unambiguous reference to that information package. This unique identifier is required so that calculations performed using package information or actions taken based on package information can be traced back to the information packages that contain the package information. In addition, this unique identifier can be used to

- Locate an information package in an information package collection
- Identify predecessor and successor information packages
- Identify related information packages
- Serve as the unique key that identifies an information package in a database

The <PackageIdentification> element contains the required package identifier, optional package identifiers for predecessor and successor information packages, optional alternate identifiers for the information package or package information, optional package description and status, and creation and modification timestamps.

4.1 UNIQUE PACKAGE IDENTIFIER

The package identifier is specified using <PackageIdentifier> element attributes. The required site attribute contains a standard NSE two-letter site identifier and the required identifier attribute contains the base identifier. The base identifier is an alphanumeric string that is not used to identify any other information package at the site. Together these attribute values form an information package identifier that is unique within the NSE.

Any process that generates a unique value can be used to generate the value for the identifier attribute. The values listed below are very likely to be unique and are acceptable values:

- Integer number that represents the system time in milliseconds plus a random 4-digit number
- Hash value generated by using a hash function such as Secure Hash Algorithm [FIPS 180-3] to generate a hash from the system name, system time, and some or all of the package information
- Unique identifier assigned by Windchill PDMLink or other product information management system to the object representing the information package

4.2 VERSIONS

The package identifier can include an optional version that can be used in situations where information already stored in information packages may be modified. This version is specified using optional <PackageIdentifier> attributes revision and instance. An information package may use one or both of these attributes. The values of these attributes may be any alphanumeric string. One possible way of using these attributes when information packages are stored in Windchill PDMLink is to store the value of the Windchill PDMLink revision in the package identifier revision attribute and the value of the Windchill PDMLink iteration in the package identifier instance attribute.

Versions can be also implemented without using the revision and instance attributes by using the <PredecessorIdentifier> and <SuccessorIdentifier> elements to specify the order of the information package versions.

4.3 PREDECESSOR AND SUCCESSOR PACKAGE IDENTIFICATION

Optional <PredecessorIdentifier> and <SuccessorIdentifier> elements identify predecessor and successor information packages. These elements have the same attributes as the <PackageIdentifier> element and are expected to have the same values as the <PackageIdentifier> element in the referenced package.

4.4 ALTERNATE IDENTIFIER

Optional <AlternateIdentifier> elements specify alternate identifiers for the information package or package information. The name attribute specifies the name of the identifier, usedFor attribute can be package or information, and the identifier is in the element content. A value of package means that the identifier is for the whole package and a value of information means the identifier is for

the package information. For example, if the whole information package was stored in a separate system and had an identifier in that system separate from the identifier in the <PackageIdentifier> element, <AlternateIdentifier> can be used to specify that other system identifier.

4.5 PACKAGE DESCRIPTION AND STATUS

Optional <PackageDescription> and <PackageStatus> elements describe the package information. Element <PackageDescription> can contain a general text description of the package information. Element <PackageStatus> can contain a lifecycle state or other phrase describing the status of the information such as active, obsolete, or released.

4.6 PACKAGE TIMESTAMPS

Optional element <CreatedTimestamp> can contain a timestamp that specifies the date and time the information package was created. Optional element <ModifiedTimestamp> can contain a timestamp that specifies the last date and time the information package was modified. The timestamp is in the element content and should conform to the timestamp format recommended in section 8 of this specification.

The specification allows for only one <ModifiedTimestamp> element in <PackageIdentification> because this timestamp is designed to be used only to help identify the information package. If an information package history must be maintained, this history should be recorded as events in the <History> element.

5. INFORMATION MARKING

All Department of Energy classified and unclassified controlled information must be protected by marking it as required by DOE regulations. These markings warn those in possession of classified or unclassified controlled information to control access to that information and to protect that information when storing it.

Element <InformationMarking> contains the information marking required to protect the information package and its package information. If the information is classified, required markings are placed in its child <Classification> element. If the information is unclassified controlled information, required markings are placed in its child <UnclassifiedControlled> element.

Information packages are expected to be used to store a wide variety of classified and unclassified information. To ensure that classified information is protected, element <Classification> is always required. The classification level must be in its child <Level> element content and its other child elements must contain the information marking required to protect the information package and package information. If the information is unclassified, the <Level> element must contain Unclassified.

If the information is unclassified, the <Classification> element can be followed by one or more <UnclassifiedControlled> elements. Each <UnclassifiedControlled> element identifies one type of unclassified controlled information present in the information package in its <Type> element and contains the information markings required to protect that unclassified controlled information type in other

child elements. If the information is unclassified and not controlled, one <UnclassifiedControlled> element must be present and contain a <Type> element with Not Controlled in its element content.

The <InformationMarking> element is the second element in the information package after the element that contains the package identification. This placement ensures visibility when the file is viewed. The package unique identifier in the <PackageIdentification> element is considered part of the markings when the information in the package is considered accountable.

5.1 CLASSIFIED INFORMATION

DOE manual 470.4-4A issued in January 2009 [DOE M 470.4-4A] specifies markings required for classified documents and material. Document marking requirements specify markings for traditional paper documents and for other human-readable media formats. Material markings require the existence of a drawing that contains required marking information. There are no requirements for other information formats such as XML documents. The only requirement is [DOE M 470.4-4A]:

> *Classified matter, regardless of date or agency of origin, must be marked to indicate at least the classification level and category (if RD or FRD).*

The classified information marking requirements in this specification are based on the document marking requirements in DOE manual 470.4-4A.

5.1.1 Classification Level and Category

Element <Level> contains the highest classification level of the information in the information package and is required. Its element content must be one of these four values:

- Top Secret
- Secret
- Confidential
- Unclassified

The capitalization used in the level is required because it is the capitalization used for these words in DOE manual 470.4-4A. To ensure that software can recognize the value, only one blank must separate the words in Top Secret.

If the information in the information package is unclassified, this element must be present and contain Unclassified as its element content. The presence of Unclassified tells users that the information package does not contain classified information.

The <Category> element specifies the highest classification category of the information in the information package. Its element content must be one of these three values:

- Restricted Data

- `Formerly Restricted Data`
- `National Security Information`

Only one blank must separate the words in the category names. The capitalization used in the category is required because it is the capitalization used for these words in DOE manual 470.4-4A.

The classification category is required if the classification category is Restricted Data or Formerly Restricted Data. It is optional if the classification category is National Security Information. It must not be present if the classification level is Unclassified.

5.1.2 Caveats and Special Control Markings

Caveats and special control markings are placed on documents to identify special handling or dissemination requirements or to assist in describing the type of information involved or who distributed or originated the information. An information package caveat is placed in the element content of a `<Caveat>` element and a special control marking is placed in the element content of a `<SpecialControlMarking>` element. If a caveat or special control marking is not required, the corresponding element should not be present. For example, an information package containing nuclear weapon information considered Sigma 1 information would have the following element (marking is for example purposes only) [CMPC Marking]:

```
<Caveat>Nuclear Weapon Data, Sigma 1</Caveat>
```

5.1.3 Admonishment

If the package information is `Restricted Data` or `Formerly Restricted Data`, the corresponding admonishment is placed in the element content of the `<Admonishment>` element. Otherwise this element should not be present.

5.1.4 Document Title, Originator, Originating Organization and Date

The information package can be marked as a formal document using the markings described in DOE manual 470.4-4A. When the information package is marked as a formal document, the document title is placed in the `<DocumentTitle>` element, document originator in the `<DocumentOriginator>` element, and the document originator organization name and address in the `<OrganizationName>` and `<OrganizationAddress>` elements. The document title must be marked as required by DOE manual 470.4-4A. The organization name and address are text strings with parts separated by commas or other delimiters as necessary.

If the `<CreatedTimestamp>` or `<ModifiedTimestamp>` elements are not present in the `<PackageIdentification>` element, the `<DocumentDate>` element can be used to specify a document preparation date. Otherwise the date in the `<CreatedTimestamp>` or `<ModifiedTimestamp>` serves as the document date. The date in `<DocumentDate>` does not have to conform to the date and time formats recommended in this specification.

5.1.5 Classification Determination

The information on how the classification was determined is placed in the <ClassifiedBy> and <DerivedFrom> elements. The element tag names are based on the corresponding classifier markings lines required on documents by DOE manuals 470.4-4A and 475.1-1B. The classifier identification is placed in <ClassifiedBy> attributes and the source of the classification guidance is placed in <DerivedFrom> element content. Formats and contents of the element content of these elements are specified by DOE manuals 470.4-4A and 475.1-1B [DOE M 470.4–4A, DOE M 475.1–1B].

Element <DateReviewed> can be used to specify the date the document was reviewed. DOE manuals 470.4-4A and 475.1-1B assume that the document date is the date reviewed. However, information packages may not be formally reviewed until long after the packages are created. Furthermore, they may be changed after the formal review is performed without another classification review being performed. The <DateReviewed> element content contains the most recent date a formal classification review was performed on the package information. A user can use this information to judge whether this classification review is valid for the package information. This date can be in any standard date format.

Element <DeclassifyOn> contains required declassification information in its element content for information packages that contain only National Security Information. The contents and format of this information is specified by DOE manuals.

5.1.6 Additional Information

Element <AdditionalInformation> is optional and can be used to provide additional information about the classification decision. The following are examples of additional information that can be stored using this element:

- Statement “Derivative Declassifier review required prior to declassification” required when the information package contains only National Security Information.
- Sources used to make the classification determination when multiple sources are used
- Classification level and category matrix when the information package contains information in multiple classification categories
- Additional information about the classification process used when an automated process is used to make the classification determination

Each independent statement should be in a separate <AdditionalInformation> element. If additional information is not required, this element should not be present.

5.2 UNCLASSIFIED CONTROLLED INFORMATION

Unclassified controlled information (UCI) is information for which disclosure, loss, misuse, alteration, or destruction could adversely affect national security or governmental interests. National security interests are those unclassified matters that relate to the national defense or foreign relations of the U.S. Government. Governmental interests are those related, but not limited to, the wide range of

government or government-derived economic, human, financial, industrial, agriculture, technological, and law-enforcement information as well as the privacy or confidentiality of personal or commercial proprietary information provided to the U.S. Government by its citizens. In addition, other unclassified sensitive information is information that, based on a determination by competent authority (e.g., information owners), may require mandatory protection because of statutory or regulatory restrictions or may require a degree of discretionary protection because inadvertent or deliberate misuse, alteration, disclosure, or destruction could adversely affect national, Department of Energy/National Nuclear Security Administration (DOE/NNSA), or DOE contractor interests. [Y19-206]

These types of unclassified controlled information may be present at the Y-12 National Security Complex:

- Unclassified Controlled Nuclear Information (UCNI)
- Export Controlled Information (ECI)
- Naval Nuclear Propulsion Information (NNPI)
- Safeguards Information (SI)
- Sensitive Nuclear Technology (SNT)
- Official Use Only (OUO)
- Applied Technology (AT)
- Cooperative Research and Development Agreement (CRADA) information
- Confidential/Foreign Government Information–Modified Handling (C/FGI-MOD)
- Privacy Act information
- Proprietary Information
- Company-owned information

Each of these types of unclassified controlled information has its own marking requirements. An information package can contain more than one of these unclassified controlled information types. When is it does and all package information is unclassified, the required markings for each type present must be included in the information marking.

If the information package contains classified information, all information markings must be in the `<Classification>` element. `<UnclassifiedControlled>` elements must not be present.

Element `<UnclassifiedControlled>` is designed to be used for all unclassified controlled information types listed above, for all unclassified controlled types not included in the list, and for any types that may be defined in the future. Each type of unclassified controlled information present in the information package must have a corresponding `<UnclassifiedControlled>` element that contains its required information markings.

The sections below describe `<UnclassifiedControlled>` child elements and then show how they are used to protect two common types of unclassified controlled information: Unclassified Controlled Nuclear Information (UCNI) and Official Use Only (OUO) information. Markings for other types of unclassified controlled information are described in Appendix C.

5.2.1 Element Descriptions

Unclassified controlled information markings are placed in these <UnclassifiedControlled> child elements:

<Type>
<Caveat>
<DocumentTitle>
<DocumentOriginator>
<OrganizationName>
<OrganizationAddress>
<DocumentDate>
<Admonishment>
<ClassifiedBy>
<NameOrganization>
<Reviewer>
<ReviewingOfficial>
<DerivedFrom>
<Guidance>
<GuidanceUsed>
<DateReviewed>
<AdditionalInformation>

These element tag names are designed to match as closely as possible the labels used in information marking requirements. In some cases elements with different tag names are used to represent basically the same information. This specification assumes that the information marking for an unclassified controlled information type will use the element with the tag name that is the closest match to the label specified in the information marking requirements.

If package information is a controlled type, the <Type> element contains the name of the controlled information type. If the type is listed above, the corresponding name below must be placed in <Type> element content:

- Unclassified Controlled Nuclear Information
- Export Controlled Information
- Naval Nuclear Propulsion Information
- Safeguards Information
- Sensitive Nuclear Technology
- Official Use Only
- Applied Technology
- Cooperative Research and Development Agreement

- Confidential/Foreign Government Information–Modified Handling
- Privacy Act Information
- Proprietary Information
- Contractor Information

Only one blank must separate the words in the type name. If the information is not controlled, the <Type> element must contain Not Controlled.

Element <Admonishment> contains the admonishment text specified by the information marking requirement and element <Caveat> contains any caveat statements required to properly mark the information.

When the information package is marked as a formal document, the document title is placed in the <DocumentTitle> element, document originator in the <DocumentOriginator> element, and the document originator organization name and address in the <OrganizationName> and <OrganizationAddress> elements. The document title must be marked as required by DOE manual 470.4-4A and the organization name and address are text strings with parts separated by commas or other delimiters as necessary. If a document date is required, it is placed in the <DocumentDate> element.

Elements <NameOrganization>, <ReviewingOfficial>, and <Reviewer> contain the identity of the person or entity that determined the unclassified controlled information type.

Element <DateReviewed> contains the date of the unclassified controlled information type was determined. This date does not have to conform to the date formats recommended in this specification and can be in any acceptable format.

The guidance used to make a determination is entered in the element content of the <Guidance>and <GuidanceUsed> elements. The content of these elements is specified by the requirements of the information type.

Elements <ClassifiedBy> and <DerivedFrom> identify the classifier and guidance used to mark Confidential/Foreign Government Information–Modified Handling information.

Element <AdditionalInformation> provides additional information about the marking.

5.2.2 Unclassified Controlled Nuclear Information (UCNI)

UCNI is unclassified government information that is prohibited from unauthorized dissemination under Section 148 of the Atomic Energy Act and further defined in Title 10, Code of Federal Regulations, Part 73. It includes the following information:

- Design of production or utilization of facilities related to atomic energy defense programs.
- Design-related operational information concerning the production, processing, or utilization of nuclear material for atomic energy defense programs.
- Physical security measures for the protection of production or utilization facilities related to atomic energy defense programs.

If the information package contains UCNI information, the package must have an <UnclassifiedControlled> element with the UCNI admonishment and reviewing official information. The <Admonishment> element must be present and contain this element content:

Unauthorized dissemination subject to civil and criminal sanctions under section 148 of the Atomic Energy Act of 1954, as amended (42 U.S.C. 2168)

Elements <ReviewingOfficial> and <GuidanceUsed> must be present and contain the information required for the reviewing official and guidance used lines by DOE manual 471.1-1 [DOE M 471.1–1]. Element <DateReviewed> contains the review date.

If the information requires a dissemination controlled marking, the following text is placed in the <Caveat> element:

DISSEMINATION CONTROLLED Distribution authorized to DOE and DOE contractors only. Other requests shall be approved by the cognizant DOE program office, which is (office name), before release.

The (office name) text is replaced by the name of the DOE program office.

5.2.3 Official Use Only (OUO) Information

Official Use Only information is information that is unclassified and meets both of the following criteria [DOE M 471.3-1]:

- Have the potential to damage governmental, commercial, or private interests if disseminated to persons who do not need the information to perform their jobs or other DOE-authorized activities.
- Fall under at least one of eight Freedom of Information Act (FOIA) exemptions (exemptions 2 through 9).

If the information package contains OUO information, the <Admonishment> element must be present and contain:

May be exempt from public release under the Freedom of Information Act (5 U.S.C. 552), exemption number and category: (number and category) Department of Energy review required before public release.

The (number and category) text must be replaced by an OUO exemption number and text listed in DOE manual 471.3-1 or its successors. The name and organization of the person that made the determination is placed in the <NameOrganization> element content, the date the determination was made is placed

in the <DateReviewed> element content, and the guidance used (if any) is placed in the <Guidance> element content.

5.3 DEFAULT CLASSIFICATION AND CONTROLLED INFORMATION DETERMINATIONS

Much of the information expected to be stored in information packages at Y-12 was collected using automated processes and stored directly in a product information management system. The process did not route the information to a derivative classifier so it never received a derivative classifier review. The information was assigned by default the highest classification level and category that the systems that created and stored it were authorized to process.

To identify information that never received a derivative classification review, the <InformationMarking> element has a required attribute named reviewed that can have a value of yes or no. A value of yes means the information package was formally reviewed by an authorized person or software package.

If the information is classified, the <Level> element must be present and contain the default classification level. If the default classification category is Restricted Data or Formerly Restricted Data, the <Category> element must contain the classification category and the <Admonishment> element must contain the appropriate admonishment. Any required caveats and special control markings must be present. An <UnclassifiedControlled> element must not be present.

If the information is unclassified, element <Classification> and its child <Level> element must be present and <Level> must contain Unclassified. If package information is not a controlled type, an <UnclassifiedControlled> element must be present and contain Not Controlled. Otherwise an <UnclassifiedControlled> element must be present for each type of controlled information present in the information package.

The <AdditionalInformation> element and the elements that identify reviewers may be used to record additional information about how the default classification level, classification category, and controlled information type were determined.

6. ACCESS CONTROL

Information marking cannot be used by itself to determine whether a person has access to package information. It can be used to determine whether the user has an access authorization (security clearance) that allows the user access to the information, but it cannot be used to determine whether a user has a need-to-know that information. Required information package <AccessControl> element contains information attributes that can be used to determine whether a user has a need-to-know the package information.

Each information attribute in <AccessControl> is stored as a name-value pair in an <InfoAttribute> element. The information attribute is specified in the <InfoAttribute> element

name attribute and the value is specified in the element content. Element <AccessControl> can contain as many <InfoAttribute> elements as needed to protect the information. This design allows the information package creator to specify the all of the information attributes required to protect the information when the information package is created. Information managers responsible for managing information packages can later modify the information attributes as needed to continue to protect the information.

An information management system is expected to use these information attributes to control access to the information. The attributes can be read when the package is stored or read when access is requested. In either case, the information management system will use the information attributes and its own set of access control rules to determine whether access is allowed or denied.

The following example shows what <AccessControl> might contain if the package information was inspection data for the fictitious W00 program. This access control element would allow access to persons involved in the W00 program and to inspection auditors:

```
<AccessControl>
    <InfoAttribute name="program">W00</InfoAttribute>
    <InfoAttribute name="role">auditor</InfoAttribute>
</AccessControl>
```

The system that controls access to the information package would have the final say on whether access was granted.

7. SEARCH TERMS

This information package specification does not specify the organization or contents of the package information. It simply states that package information is contained in one or more <PackageInfo> element child elements. A user using an information management system or database that provides users with the capability to search XML elements and attributes can use these capabilities to locate information packages that contain specific information. However, if that user does not exactly specify in the search the elements that contain the search information, the search will fail. To specify the exact elements, a user must know the detailed structure of all information packages being searched.

The information package can contain an element named <SearchTerms> that contains a set of search terms that describe the package information. Placing all search terms in this single location makes it easy for users searching for information to specify the locations of the search terms in all information packages being searched. All search terms are stored in <SearchTerm> elements in the <SearchTerms> element content. The <SearchTerms> element can be empty or can contain as many <SearchTerm> elements as needed to enable users to locate package information.

Each <SearchTerm> element contains the search term in its element content and optional attributes that specify the name of the search term and the units of measure used for the search term. These attributes allow the user to further limit the scope of the search for the search terms.

8. INFORMATION REQUIREMENTS

Information packages may be stored for decades so the information in them should be stored using formats that will remain understandable for decades. Only formats that conform to recognized standards such as those described in this section can be expected to be understandable for this length of time. In addition, package information should not include proprietary information, undocumented formatted information, copyright or copy-protected information, or references to external entities.

8.1 DATE AND TIME FORMATS

International Organization for Standardization standard 8601 [ISO 8601] specifies standard numerical representations for dates, times and combined dates and times. A note submitted to the World Wide Web Consortium recommended that a subset of these formats sufficient to satisfy most requirements be defined and used in web applications [W3C Datetime]. This format was subsequently adopted as the format for XML Schema simple type xsd:dateTime that represents timestamps [XML Schema]. When this simple type is used to define a timestamp stored in an attribute or element, XML Schema can validate the timestamp and report invalid timestamps.

The following description of this timestamp format was extracted from [W3C Datetime].

The formats are as follows. Exactly the components shown here must be present, with exactly this punctuation. Note that the "T" appears literally in the string, to indicate the beginning of the time component, as specified in ISO 8601.

Year:
YYYY (e.g., "1997")
Year and month:
YYYY-MM (e.g., "1997-07")
Complete date:
YYYY-MM-DD (e.g., "1997-07-16")
Complete date plus hours and minutes:
YYYY-MM-DDThh:mmTZD (e.g., "1997-07-16T19:20+01:00")
Complete date plus hours, minutes and seconds:
YYYY-MM-DDThh:mm:ssTZD (e.g., "1997-07-16T19:20:30+01:00")
Complete date plus hours, minutes, seconds and a decimal fraction of a second
YYYY-MM-DDThh:mm:ss.sTZD (e.g., "1997-07-16T19:20:30.45+01:00")

where:

```
YYYY = four-digit year
MM  = two-digit month (01=January,   etc.)
DD  = two-digit day of month (01 through   31)
hh  = two digits of hour (00 through 23)   (am/pm NOT allowed)
mm  = two digits of minute (00 through   59)
```

```
ss  = two digits of second (00 through   59)
s   = one or more digits representing a decimal fraction of a second
TZD = time zone designator (Z or +hh:mm or   -hh:mm)
```

This profile defines two ways of handling time zone offsets:

1. Times are expressed in UTC (Coordinated Universal Time), with a special UTC designator ("Z").
2. Times are expressed in local time, together with a time zone offset in hours and minutes. A time zone offset of "+hh:mm" indicates that the date/time uses a local time zone which is "hh" hours and "mm" minutes ahead of UTC. A time zone offset of "-hh:mm" indicates that the date/time uses a local time zone which is "hh" hours and "mm" minutes behind UTC.

If another date or time format is used, it should clearly identify whether it is based on coordinated universal time (UTC, also known as Greenwich Mean Time or GMT) or is a local time. If it is a local time, the difference between the local time and UTC should be specified. It can be specified using a time zone such as Eastern Daylight Time (EDT) or by an offset from UTC such as -04:00 (EDT) or -05:00 (Eastern Standard Time).

8.2 MEASUREMENT UNIT ABBREVIATIONS

The abbreviations used for measurement units defined by *The International System of Units* [SI] should be the abbreviations defined by that standard. The abbreviations used for traditional US units of measurement such as inch, pound, or gallon should follow the conventions specified in National Institute for Standards and Technology (NIST) handbook 44 [NIST 44].

8.3 CHARACTERS & AND < IN ELEMENT CONTENT

The XML standard prohibits the use of & and < characters in element content except when those characters are used to identify an entity (&) or the start of a tag (<). XML fragments, text, and data that contains & or < cannot be stored in element content without protecting the text by taking one of the actions described below. Characters & and < are allowed in attribute content.

The standard provides two methods for including & and < characters in element content. The & character can be replaced by its entity reference & and the < character can be replaced by its entity reference <. The recommended alternative is to insert <![CDATA[before the start of the text that contains & and/or < and place]]> at the end of the text. An XML processor considers all characters between <![CDATA[and]]> to be a single block of text and performs no processing on it.

8.4 BINARY INFORMATION

An XML document is designed to store character information. Information not in character form must be converted to a character form before it can be stored in an XML document. Internet Engineering Task Force Request for Comments 4648 *The Base16, Base32, and Base64 Data Encodings* [RFC 4648] specifies standard methods for converting binary content to and from character encodings suitable for use in XML documents. The base 64 encoding defined in this RFC should be used to convert binary data and other data not compatible with XML to a character-based encoded form before that information is stored in

an information package. When converting a base 64 text encoding to binary data, characters outside the base encoding alphabet must be ignored when interpreting data. The characters that represent encoded binary information may include white space characters such as spaces, tabs, carriage returns, and line feeds when required to make the encoding easier to read in an XML document.

The only types of binary information stored in an information package should be types that can be expected to be useful for the expected life of the information package. These types should be well-recognized standardized types such as Portable Document Format (PDF) and graphics interchange format (GIF).

Information in a proprietary binary format stored in an information package may not be usable for the life of an information package because software required to process that information is no longer available. If possible, a non-proprietary version of the information should be included in the information package to ensure that the information is preserved after the software is no longer available.

The element that contains the binary encoding content should describe the format of the contained information or provide sufficient information for the content to be usable by a user unfamiliar with the contents of the information package. For example, if the contained binary information is a Portable Document Format (PDF) document stored using base 64 text encoding, the element that contains the PDF document binary encoding should specify that the contents represent a PDF document and was encoded using base 64.

8.5 FORMATTED INFORMATION

Formatted or other structured information may be stored in an information package. This formatted or structured information should either include a description of the information sufficient for a person to understand the information for the expected life of the information package or a reference to information expected to be available for the life of the information package.

8.6 COPYRIGHTED INFORMATION

An information package submitted to an archive may be provided to many persons at different NSE sites. If the information package contains copyrighted or licensed information or has copy restrictions, providing the information package to others may result in a violation of the copyright or license agreement. Copyrighted and information with copy restrictions should not be stored in information packages.

8.7 EXTERNAL INFORMATION

Information packages should not contain references to external entities such as XML entities or files or other objects not stored using information packages. The only XML entities that can safely be used are those defined by the XML standard. Any reference to an external file is likely to be lost over time.

9. XML SIGNATURES

A digital signature implemented as an XML signature is used to detect changes to package information and to confirm the origin of the information. A digital signature is described in National Institute of Standard and Technology (NIST) Federal Information Processing Standards (FIPS) *Digital Signature Standard* (DSS) as follows [FIPS 186-3]:

> *A digital signature is an electronic analogue of a written signature; the digital signature can be used to provide assurance that the claimed signatory signed the information. In addition, a digital signature may be used to detect whether or not the information was modified after it was signed (i.e., to detect the integrity of the signed data). These assurances may be obtained whether the data was received in a transmission or retrieved from storage.*

W3C *XML Signature* recommendation [XMLDSIG 2002] and its second edition [XMLDSIG 2008] specify XML syntax and processing rules for creating and representing digital signatures in XML documents. These signature recommendations are an implementation of the NIST FIPS 186-3 *Digital Signature Standard* for XML documents.

Packaged information can be protected by one or more digital signatures that conform to the first W3C XML signature recommendation [XMLDSIG 2002], the second edition of that recommendation [XMLDSIG 2008], or a subsequent edition, revision, or replacement of that recommendation. The revision or edition of the standard used will be identified in the signature as required by the standard.

9.1 DIGITAL SIGNATURES

A digital signature is a string of bits that represents both the signed content and the signer. The signed content is represented by a message digest created by using a hash function to transform the content to a string of bits. This string of bits is encrypted by the signer using a standard encryption algorithm and key known to the signer. The bit string generated by the encryption algorithm is the digital signature. When public key cryptography is used, the signer's private key is used to encrypt the bit string.

The signature can be verified by using the same hash function to create a message digest from the signed content. When the signer and verifier share a secret key, the verifier uses the same encryption algorithm and key to decrypt the message digest. When public key cryptography is used, the verifier uses the signer's public key to decrypt the message digest. If message digest calculated by the verifier matches the message digest in the message, the signatures match and the verifier knows that the signed content was signed by the signer and has not changed since the signature was calculated.

Digital signatures are secure because of the characteristics of the hash and encryption functions used to create them. A hash function algorithm is considered secure if, for a given algorithm, it is computationally infeasible (1) to find a message that corresponds to a given message digest, or (2) to find two different messages that produce the same message digest. Any change to a message will, with a very

high probability, result in a different message digest [FIPS 180-3]. The Secure Hash Algorithm (SHA) specified by [FIPS 180-3] and used in XML signatures meets these requirements.

The message digest is encrypted by the signer to show that the signer had access to the key required to encrypt it. If a secure encryption algorithm was used, it is computationally infeasible to create the appropriate signature bit string without knowing the encryption key. Since encryption is reversible, every unique encrypted bit string has one corresponding decrypted bit string and vice versa. Together these characteristics ensure that, if the decrypted signed message digest in the message corresponds to the message digest calculated from the message, the signer had access to the key used to sign it and the message has not changed since it was signed.

A message can be signed using a shared secret key or by using public and private keys. A shared secret key is shared by the signer, the verifier, and possibly by others. The key must be protected from unauthorized users and uses and it must be securely preserved for the life of the information packages it was used to sign. These characteristics make it impractical to use shared secret keys to sign information packages.

Public-key cryptography uses asymmetric key algorithms instead of the symmetric key algorithms used with shared secret keys. Asymmetric key algorithms use a pair of keys instead of the single shared key used in symmetric algorithms. If the two keys are designated *k1* and *k2*, then using the appropriate asymmetric key algorithm, a message encrypted using *k1* must be decrypted using *k2* and a message encrypted using *k2* must be decrypted using *k1*. Furthermore, it is computationally infeasible to use one of the keys in a key pair to determine the other key. This allows a user to keep key *k1* private and allow key *k2* to be publicly known. A person can digitally sign a message by using a private key to encrypt the message digest. A second person can verify the digital signature by decrypting the signed message digest using the corresponding public key, computing the message digest, and comparing the two digests. If they match, a person in possession of the private key signed the message and it has not changed since it was signed.

These characteristics allow digital signatures to be used for these three purposes:

- Detect unauthorized modifications to signed information
- Authenticate the identity of the signer
- Prevent the signer from later repudiating the signature and signed information

If a message is altered in any way without signing the information again, the secure hash algorithm will calculate a significantly different message digest and an attempt to verify the signature will fail. A verification of a digital signature using a public-key cryptography public key proves that the corresponding private key was used to sign the message and that the message came from a source that knew the private key.

9.2 DIGITAL SIGNATURES IN INFORMATION PACKAGES

Digital signatures are used in information packages primarily to detect changes to protected information stored in the packages. These changes are most likely to occur as a result of accidental damage

to the information. Such damage can be the result of transmission errors when moving an information package or from damage to the medium used to store the information package.

Digital signatures also allow analysts using an information package to verify that the information in the package is the same information that was stored in the package when it was created. The digital signature allows the analyst to eliminate information from an analysis where digital signatures show that the information package that contains the information has changed and the information may not be valid.

The digital signature also authenticates the origin of the information. This factor is important only if the information in an information package may be deliberately modified or if the origin of the information is uncertain. Every digital signature consists of a message digest encrypted using a private key. The corresponding public key is encoded in a certificate that is included in the digital signature. Each certificate has an associated value called a fingerprint that is unique to the certificate. If the fingerprints calculated from two certificates are identical, the certificates are identical. If the signed contents of an information package were deliberately altered and were signed, the signing certificate will be different unless the signer had access to the private key used to sign the original signed contents. This different certificate will be revealed by comparing its fingerprint to the fingerprint obtained from certificates in valid information packages. Likewise, the source of an information package can be determined by comparing the fingerprint of its certificate to the fingerprints of certificates from information packages with known sources.

9.3 XML SIGNATURES

XML signature standards [XMLDSIG 2002, XMLDSIG 2008] and the references specified in these standards describe the digital signature standard for XML documents. These specifications have been implemented in a number of software packages. This information package specification assumes that a software package implementation that conforms to these standards will be used to sign the signed contents of information packages, so a discussion of how digital signatures are created is not appropriate. The XML signature specifications do specify a number of options and these options are discussed in this section.

The XML signature implemented in this specification has this implementation:

```
<Signature xmlns="XML signature namespace URL">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="canonicalization method URL"/>
    <SignatureMethod Algorithm="signature method URL"/>
    <Reference URI="#SignedContents">
      <Transforms>
        <Transform Algorithm="XML signature enveloped signature URL "/>
      </Transforms>
      <DigestMethod Algorithm="XML signature digest method URL"/>
      <DigestValue>Dhg ... XU=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>Dc ... Oc=</SignatureValue>
  <KeyInfo>
     <X509Data>
       <X509SubjectName>CN= ... ,C=US</X509SubjectName>
```

```
        <X509Certificate>MIIE ... Zrq8=</X509Certificate>
      </X509Data>
    </KeyInfo>
  </Signature>
```

These elements have the following meanings:

Element Tag Name	Element Description
Signature	Container for the XML signature. The xmlns attribute sets the namespace to the XML signature namespace to ensure that child elements are considered XML signature elements and not elements in other namespaces.
SignedInfo	Container for the information to be signed. Its child elements specify the message digest and how it is created and signed.
CanonicalizationMethod	The method used to convert the signed contents to its canonical form.
SignatureMethod	The name that represents all methods used to create the signature. It includes the name of the digest method and the name of the method used to sign the digest value.
Reference	Identifies the information to be signed. Its URI attribute must be a document fragment that references the value of the id attribute of the signed element, either the <PackageInfo> element or one of its child elements.
Transforms	Container for <Transform> elements.
Transform	Specifies how the signed information element is processed to create the signature.
DigestMethod	The method used to calculate the message digest from the signed contents.
DigestValue	The message digest calculated from the signed contents in base 64 encoding.
SignatureValue	The message digest encrypted using the private key in base 64 encoding.
KeyInfo	Container for the certificate that contains the public key used to decrypt the message digest.
X509Data	Container for the certificate.
X509SubjectName	Certificate name.
X509Certificate	Certificate in base 64 encoding.

<CanonicalizationMethod> element Algorithm attribute defines the method used to transform the contents of the <SignedInfo> element to a standard form. Secure Hash Algorithm and other message digest algorithms calculate their digests from all of the information provided them, so a message digest value calculated from an XML document fragment will depend on the document contents and structure. W3C has issued recommendation Canonical XML versions 1.0 and 1.1 that specify a standard structure for XML documents. The goal of these specifications are to establish a method for determining whether two XML documents are identical, or whether an application has not changed a document, except for transformations permitted by XML 1.0 and Namespaces in XML 1.0. [C14N10, C14N11].

Element <SignatureMethod> defines the combination of the message digest algorithm used to create a message digest of the contents of the <SignedInfo> element and the algorithm used to encrypt it

to create the signature value. These combinations use Secure Hash Algorithm to create the message digests but differ in how the message digest is signed:

- HMAC-SHA-1
- DSAwithSHA-1
- RSAwithSHA-1

HMAC-SHA-1 uses a message authentication code encrypted using a shared secret key to create the signature. A shared secret key is not likely to be preserved over time, so this alternative is not considered acceptable.

DSAwithSHA-1 and RSAwithSHA-1 both use public key encryption to encrypt the message digest creating the signature. DSAwithSHA-1 uses the Digital Security Algorithm defined in [FIPS 186-3] and RSAwithSHA-1 uses the RSA algorithm defined in IETF RFC 2437. Algorithm DSAwithSHA-1 is required and RSAwithSHA-1 is recommended. Information packages will use DSA-based algorithms both because they are required and because it is a Federal standard.

The Java software used to sign the information packages used to develop this specification automatically converts the signed package information to a canonical form before calculating the signature. This signed package information can change as long as the change does not significantly alter the meaning of the signed contents and the signature will still be valid. For example, the order of attributes in an element is not significant in XML so it can change without invalidating the signature. Any change that alters the meaning of the signed package information such as adding or deleting elements or attributes or changing attribute or element content will invalidate the signature.

9.4 SECURITY ISSUES

The XML Signature standard specifies that secure hash algorithm SHA-1 be used to compute the message digests. However, SHA-1 is no longer considered acceptable by NIST for protecting important sensitive unclassified information. Stronger versions of SHA are available and must be used. However, until these stronger versions become part of the XML Signature recommendation, SHA-1 will be used to compute the message digests for information packages. The risks of using SHA-1 are low in this case because a malefactor is unlikely to benefit from altering an information package. The primary threat to information packages is accidental damage, and SHA-1 is capable of detecting this.

When computing a DSA public-private key pair, the key bit length must be either 1024 or 2048 [FIPS 183-3].

10. ELEMENT NAMES AND NAMESPACES

Every element in an XML document must have a unique name and definition. Two elements cannot share a name and have different definitions. For example, the `<Reference>` element defined by this specification

and the <Reference> element defined by the XML Signature specification have different definitions. An element in an information type structure cannot have the same name as an element defined by this specification or the XML Signature specification.

XML element name conflicts are resolved by placing elements in different namespaces. A namespace consists of a uniform resource identifier (URI) name and an optional prefix. A namespace and associated prefix are defined by adding this attribute to an element:

```
xmlns:pf="URI"
```

where pf is the prefix and URI is the namespace uniform resource identifier. The prefix can be an empty string. In this case, the namespace definition attribute is

```
xmlns="URI"
```

The namespace definition applies to the element that contains the definition and all elements below it in the XML structure. An element is added to a defined namespace with a prefix by adding the prefix to the element name. When ip is defined as the prefix for the information package namespace, the information package document root element name becomes <ip:InfoPackage>. When the information package namespace is defined using an empty string prefix, the document root element name remains <InfoPackage>. The colon is not used to define namespace with an empty string prefix or to add an element to the namespace.

A namespace must be defined in the element at the base of the XML structure that uses the namespace or at a higher level. For example, all namespaces can be defined in the document root element, each namespace can be defined in the element at the base of the XML structure that uses the namespace, or a combination of the two approaches can be used.

The namespace associated with a prefix (including the empty string prefix) can be changed by adding the xmlns definition of that namespace to an element. This new definition applies to that element and to all elements below it in the XML structure. For example, the information package namespace with the empty string prefix can be used for all elements defined by this specification. A package information namespace can be defined using the empty string prefix and its definition added to all child elements of <PackageInfo> that store package information. The XML Signature specification <Signature> element defines its own namespace using the empty string prefix. In this way, all element definitions are unique (including the two <Reference> elements) with no prefixes used.

Information packages that use namespaces can only be defined and validated using XML Schema (XSD) definitions. XML Schema was developed after namespaces were defined and incorporate namespaces in their definitions. The XML document type definition (DTD) was part of the original XML specification so it does not recognize namespaces. Namespaces can be simulated in DTDs by adding the namespace definition xmlns attribute to the element definition and by adding the namespace prefix to the names of the elements in the namespace. In practice, all this does is change the names of the elements.

If a DTD is used to define and validate an information package, this specification recommends that the `ds:` prefix be used for all XML Signature elements. This is easily done as these elements are typically generated by software and the namespace prefix is usually one of the available software settings. XML Signature element definitions in a DTD must include the `ds:` prefix in their element names.

10.1 NAMESPACE NAMES

All information packages based on this specification should use namespace names that conform to the namespace name structure specified in this section. The Namespaces in XML version 1.1 specification states that every namespace name must be a Internationalized Resource Identifier (IRI) as described in RFC 3987 [XML Namespaces]. An IRI is an internationalized form of the Uniform Resource Identifier (URI) that allows characters from character sets other than US-ASCII to be used in identifiers [RFC 3987]. All namespace names used to specify information package namespaces should contain characters from the US-ASCII character set and should therefore conform to the URI specification in RFC 3986 [RFC 3986].

The URI specification defines a name structure that allows conforming URI names to have a number of formats. The best known conforming format is the Uniform Resource Locator (URL) used in web addresses. The XML Signature namespace uses a web URL to identify its namespace. However, using a URL as a namespace gives the impression that an XML Schema document definition can be found at that location even though the namespace standard explicitly states that the location does not have to exist and no conforming implementation should attempt to locate it.

A second acceptable URI is the Uniform Resource Name (URN) format defined in RFC 2141 [RFC 2141]. This format consist of

```
urn:namespace identifier:namespace-specific string
```

where `namespace identifier` is a string that identifies the namespace and `namespace-specific string` is a string whose meaning depends on `namespace identifier`. RFC 3406 requires that namespace identifiers be registered with the Internet Assigned Names Authority (IANA) but it allows names starting with `x-`to be used for internal or experimental purposes without registration [RFC 3406]. Registered names can be expected to be unique but names starting with `x-` are not.

This specification recommends that information packages use the namespace identifier

```
x-y12.doe.gov
```

to show that the namespace is an internal namespace for the Y-12 National Security Complex. This name conforms to the namespace identifier format requirements of RFC 2141. Other sites may substitute their unclassified domain name for the Y-12 domain name in package information namespaces.

This specification recommends that the namespace-specific string start with

```
InfoPackage:base element name:version series
```

where `InfoPackage` is a text string, `base element name` is the name of the element at the base of the XML structure that uses the namespace, and `version series` identifies the XML document definition version series. For example, the namespace used for the document root and package metadata elements defined by this specification is

```
urn:x-y12.doe.gov:InfoPackage:InfoPackage:1.1
```

If package information were stored in a `<PackageInfo>` child element named `<ProductInfo>`, the namespace used for the `<ProductInfo>` structure would be (assuming it's the first version)

```
urn:x-y12.doe.gov:InfoPackage:ProductInfo:1.0
```

These namespaces are used in the working example.

The version series identifies compatible XML document definition versions. Each XML document definition is assigned a version. Every information package XML document that uses this XML document version stores the version in the `version` attribute of the element at the base of the XML document structure. This version is also used as the version series value. If the XML document definition is revised but still defines XML documents based on the previous revision, the version associated with the XML document definition changes but the version series value does not. This allows the revised XML document definition to be used to define XML documents associated with the previous version. For example, if an XML document definition is revised by adding an optional elements or attribute, the DTD or XSD that defines the new XML document will continue to define and validate XML documents based on the previous XML document definition. In this case, the new version is added to the appropriate version attribute definition in the revised XSD file (if used) but the version series is not changed. If an incompatible change is made such as converting an optional attribute to a required attribute, the version series is set to the new version to prevent the new XML document definition from being used to define information packages based on previous XML document definitions.

10.2 NAMESPACE PREFIXES

This specification recommends that namespace prefix `ip:` be used when necessary to identify elements defined by this specification. The XML Signature specification recommends that the namespace prefix `ds:` be used to identify elements it defines. If a namespace prefix is used for `<PackageInfo>` child elements, that prefix must not conflict with the prefixes used for elements defined by this specification or the XML Signature specification.

11. INFORMATION PACKAGE SPECIFICATION

Information package document root, package metadata, and package information parent elements and associated attributes are formally defined and described in this section. Document root child elements

are defined and described in the order they appear in the information package XML document. Each document root child element definition and description includes the definitions and descriptions of all elements below it in the XML document structure.

An XML element consists of an element start tag, element content, and an element end tag. XML element start tag can contain attributes that provide more information about the element. Element content includes everything between the end of the element start tag and the beginning of the element end tag and can be other elements, text, and characters considered to be white space. Element content can be empty which means that elements and text characters not considered to be white space are not allowed in the content. Only characters considered to be white space are allowed in an element with empty content. An element with empty content can be represented by a single XML empty element tag. The XSD definitions and equivalent document type definition (DTD) definitions in the table below are used in element definition tables to specify required element content.

XSD Definition	DTD Definition	Contents and Format
Elements	Elements	Element content is the elements listed in the Element Description and Content column in the specified order. Text and characters not considered to be white space characters are prohibited.
Empty	EMPTY	Element content is empty. The element can be represented by a single empty element tag .
xsd:dateTime	PCDATA	Timestamp in the recommended timestamp format specified in section 8. If another timestamp format is used, XSD simple type xsd:token should be used to define the content.
xsd:normalized-String	PCDATA	Each occurrence of a tab, line feed, and carriage return is replaced by a space.
xsd:string	PCDATA	String stored without modification.
xsd:token	PCDATA	String in which all consecutive white space characters are replaced with a single space character and all leading and trailing white space characters are removed.

Element definition tables include a column labeled Attr? (for attributes). If the element has attributes, this column contains Yes and the attributes are defined in the next table. A value of No means the element has no attributes.

Each element attribute is a name-value pair in the form name=value with value a quoted string. Attributes can be required or optional. The XSD definitions and equivalent document type definition (DTD) definitions in the table below are used in attribute definition tables to specify attributes.

XSD Definition	DTD Definition	Contents and Format
xsd:dateTime	CDATA	Timestamp in the recommended timestamp format specified in section 8. If another timestamp format is used, XSD simple type xsd:token should be used to define the content.
xsd:string	CDATA	String stored as provided by the process that stores the string.
xsd:token	CDATA	String in which all consecutive white space characters are replaced with a single space character and all leading and trailing white space characters removed.

Attribute definition tables include a column labeled Reqd? (for required). A value of Req in this column means the attribute is required and a value of Opt means it is optional.

Namespace prefixes are not used in element names and the namespace `xmlns` attribute definitions are not included. These attribute definitions may be provided on the `<InfoPackage>` document root element and `<PackageInfo>` child elements.

11.1 ELEMENT `<InfoPackage>`

Element `<InfoPackage>` is the document root element of the information package document. Its child elements are the parent elements of the package metadata elements and the parent element of the package information XML structure.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
`InfoPackage`	Elements	Yes	Information package document root element. Element content is One `<PackageIdentification>` element One `<InformationMarking>`element One `<AccessControl>`element Zero or one `<SearchTerms>`elements Zero or one `<References>`elements Zero or one `<History>`elements Zero or one `<Notes>`elements One `<PackageInfo>`element

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
`version`	xsd:token	Req	Information package specification version. For information packages that conform to this specification, the attribute value must be "`1.1`".

The `version` attribute describes the structure and contents of the information package. The value of `1.1` tells users and software applications that the information package consists of this document root element and the package metadata elements and information structure elements described in this specification.

If the information package and package metadata elements are in a namespace, the `xmlns` namespace definition attribute must be present in the element start tag and it must specify a namespace such as

```
urn:x-y12.doe.gov:InfoPackage:InfoPackage:1.1
```

If the `xmlns` attribute defines a namespace prefix, this prefix must be added to the names of the document root `<InfoPackage>` element, all package metadata elements, and the `<PackageInfo>` element. All package information elements must be in a different namespace and use a different namespace prefix. These namespaces should be defined on the package information elements that are child elements of the `<PackageInfo>` element. They can be also defined on this `<InfoPackage>` element but this is not recommended. The recommended prefix for information package elements is `ip`.

11.2 ELEMENT <PackageIdentification>

Element `<PackageIdentification>` contains the package metadata elements that identify the information package. Its child elements specify the unique identifier that identifies the information package, identifiers of any predecessor and/or successor information packages, alternate identifiers for the information package or the package information, an optional package description and status, and optional created and modified timestamps.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
`PackageIdentifiers`	Elements	No	Package identification elements. Element content is One `<PackageIdentifier>` element Zero or one `<PredecessorIdentifier>` elements Zero or more `<SuccessorIdentifier>` elements Zero or more `<AlternateIdentifier>` elements Zero or one `<PackageDescription>` elements Zero or one `<PackageStatus>` elements Zero or one `<CreatedTimestamp>` elements Zero or one `<ModifiedTimestamp>` elements

11.2.1 Elements <PackageIdentifier>, <PredecessorIdentifier>, and <SuccessorIdentifier>

Element `<PackageIdentifier>` contains the information package identifier and is required. The information package identifier consists of a site, base identifier, revision, and instance and are stored in these elements using attributes. The site and base identifier are required and the revision and instance are optional. When the revision and instance are specified, they are included in the information package identifier and used to ensure that the information package identifier is unique. For example, two information packages can have the same values for their site and base identifier attributes provided they do not have the same values for their revision and/or instance attributes.

When the information package is one of a series of information packages identified by versions, element <PredecessorIdentifier> can be used to identify the immediately preceding information package in the series and one or more <SuccessorIdentifier> elements can be used to identify succeeding information packages. These elements have the same attributes as the <PredecessorIdentifier> element and contain the attribute values of the referenced information package. For example, if two information packages in a series, the attribute values of the second package <PredecessorIdentifier> element will match the attribute values of the first package <PackageIdentifier> element.

Element Tag Name	XSD Definition	Attr?	Element Description and Contents
PackageIdentifier	Empty	Yes	Package identifier
PredecessorIdentifier	Empty	Yes	Package identifier of the predecessor information package
SuccessorIdentifier	Empty	Yes	Package identifier of a successor information package

Attribute Name	XSD Definition	Reqd	Attribute Description and Content
site	xsd:token	Req	NSE two-letter site identifier
identifier	xsd:token	Req	Information package base identifier
revision	xsd:token	Opt	Information package revision
instance	xsd:token	Opt	Information package instance

11.2.2 Element <AlternateIdentifier>

Element <AlternateIdentifier> specifies optional alternate identifiers for the information package or the package information. For example, if the package information was obtained from another system, element <AlternateIdentifier> can be used to record the identifier of the information in that system.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
AlternateIdentifier	xsd:token	Yes	Alternate identifier for the information package or package information. Element content is the alternate identifier.

Attribute Name	XSD Definition	Reqd	Attribute Description and Content
name	xsd:token	Req	Alternate identifier name
usedFor	xsd:token	Req	What the identifier was used for. It must be package, information, other, or unknown

Attribute name specifies the name of the alternate identifier and attribute usedFor specifies what the alternate identifier is for. It must have the value package if the alternate identifier is for the information package, information if the alternate identifier is for the package information, other if the alternate

identifier use is not `package` or `information`, or `unknown` if the use of the alternate identifier is not known. The alternate identifier is in the element content.

11.2.3 Elements <PackageDescription> and <PackageStatus>

Elements <PackageDescription> and <PackageStatus> are optional and contain text strings that describe the contents of the information package. Element <PackageDescription> can contain a general description of the package information. Element <PackageStatus> can contain a simple phrase that describes the status of the information package or package information such as released, approved, or obsolete.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
PackageDescription	xsd:string	No	Description of the information package or package information.
PackageStatus	xsd:token	No	Status of the information package or package information.

11.2.4 Elements <CreatedTimestamp> and <ModifiedTimestamp>

Element <CreatedTimestamp> can contain a timestamp that encodes the date and time the information package was created. Element <ModifiedTimestamp> can contain a timestamp that encodes the date and time the information package was last updated. The timestamps defined below assume the timestamp format recommended in section 8 is used. If a different timestamp is used, the XSD definition of this timestamp should be xsd:token.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
CreatedTimestamp	xsd:dateTime	No	Date and time the information package was created encoded as a timestamp. Element content is the timestamp.
ModifiedTimestamp	xsd:dateTime	No	Date and time the information package was last modified encoded as a timestamp. Element content is the timestamp.

11.3 ELEMENT <InformationMarking>

Element <InformationMarking> and its child elements <Classification> and <UnclassifiedControlled> contain the information marking required to protect the package information. The element design assumes that most information stored in information packages will be information that is either classified or must be documented as unclassified. Accordingly, element <Classification> is required both for classified and unclassified information. Element <Classification> for unclassified information documents the determination that the information package and its contents are unclassified. This determination may be made by a specific reviewer or review process or may be because the information came from a system not approved to process classified information.

<UnclassifiedControlled> elements are used to contain the identification of and markings for all unclassified controlled information in an information package. Each <UnclassifiedControlled>

element in an information package contains the identification of and markings for one type of unclassified controlled information present in the information package. When multiple types of controlled information are present in the information package, DOE marking rules require that markings for all types present must be used. In this case, an<UnclassifiedControlled> element must be present for each type of unclassified controlled package information. If the information is not controlled, an <UnclassifiedControlled> element must be present to document this determination. Unclassified controlled information markings cannot be used when classified information are present so an <UnclassifiedControlled> element must not be present when the information package contains classified information.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
InformationMarking	Elements	Yes	Information markings. Element content is One <Classification> element Zero or more <UnclassifiedControlled> elements Zero or more <AdditionalInformation> elements

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
reviewed	xsd:token	Req	Whether the classification and unclassified controlled information determinations were made using an approved information review process. The attribute value must be either yesor no

Attribute reviewed documents whether the classification and unclassified controlled information determinations were set by default or are the result of an approved information review process. A value of yes means the information classification or type was determined using an approved information review process. If the information package was produced on a classified system, an attribute value of no means the classification level and category were set to the highest level and category the system is authorized to process. If the information package was produced on an unclassified system, a value of no means the information type was set to a default type specified in the system documentation.

11.3.1 Element <Classification>

This element contains the determination that the information package contains classified information and the information marking required to protect the information or documentation of a determination that the package information is not classified. Its child elements identify the information classification level, category, and classifier and contain all required admonitory notices.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
`Classification`	Elements	No	Information required for classified information. Element content is One `<Level>` element Zero or one `<Category>` element Zero or more `<Caveat>` elements Zero or more `<SpecialControlMarking>` elements Zero or one `<DocumentTitle>` elements Zero or one `<DocumentOriginator>` elements Zero or one `<OrganizationName>` elements Zero or one `<OrganizationAddress>` elements Zero or one `<DocumentDate>` elements Zero or one `<Admonishment>` elements Zero or one `<ClassifiedBy>` elements Zero or one `<DerivedFrom>` elements Zero or one `<DeclassifyOn>` elements Zero or one `<DateReviewed>` elements Zero or more `<AdditionalInformation>` elements

11.3.1.1 Elements `<Level>` and `<Category>`

These elements specify the classification level and category of the information package.

Element Tag Name	XSD Definition	Attr?	Element Description and Content

`Level`	xsd:token	No	Classification level. Element content is one of these values using the capitalization and word spacing shown:
			`Top Secret`
			`Secret`
			`Confidential`
			`Unclassified`

`Category`	xsd:token	No	Classification category. Element content is one of these values using the capitalization and word spacing shown:
			`Restricted Data`
			`Formerly Restricted Data`
			`National Security Information`

The element content must use exactly the word capitalization and spacing shown with no leading or training blanks. This word capitalization is the capitalization used for these words in DOE manual 470.4-4A. Following a consistent format makes it easier for an application to recognize these terms and use them to protect the package information.

If the package information is unclassified, the `<Level>` element must be present and contain `Unclassified`. The presence of `Unclassified` shows that the package information was determined

to be unclassified by a specific reviewer or review process or was created on a system not approved to process classified information.

If the package information is Restricted Data or Formerly Restricted Data, the <Category> element must be present and contain the appropriate phrase. Element <Category> is optional if the classification category is National Security Information and must not be present if the classification level is Unclassified.

11.3.1.2 Elements <Caveat> and <SpecialControlMarking>

Elements <Caveat> and <SpecialControlMarking> identify special handling or dissemination requirements or assist in describing the type of information involved or who distributed or originated the information. Examples include Sigma level markings and the NOFORN caveat limiting distribution to foreign entities. See DOE manual DOE manual 470.4-4A for information on the contents required for these elements.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
Caveat	xsd:token	No	Caveat. Element content is the caveat.
SpecialControlMarking	xsd:token	No	Special control marking. Element content is the special control marking.

A <Caveat> element contains a single caveat and a <SpecialControlMarking> element contains a single special control marking. When multiple caveats or special control markings are required, a <Caveat> element is present for each caveat required and a <SpecialControlMarking> element is present for each required special control marking.

The phrases used for caveats and special control markings should match as closely as possible the phrases used in information marking guidance. Using consistent formats for these phrases will make it easier to design applications to use these phrases to protect the package informations.

11.3.1.3 Element <Admonishment>

An admonishment statement is a statement that warns persons with access to the information about the consequences of releasing the information to unauthorized persons. An admonishment statement is required if the information package contains Restricted Data or Formerly Restricted Data. National Security Information and unclassified information do not have admonishment statements, so this element should not be present if Restricted Data or Formerly Restricted Data are not present in the information package.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
Admonishment	xsd:string	No	Admonishment. Element content is the admonishment statement.

If the information package contains Restricted Data, the following text must be in the <Admonishment> element content:

This document contains Restricted Data as defined in the Atomic Energy Act of 1954, as amended. Unauthorized disclosure is subject to Administrative and Criminal Sanctions.

If the information package contains Formerly Restricted Data but no Restricted Data, the following text must be in the <Admonishment> element content:

Unauthorized disclosure subject to Administrative and criminal sanctions. Handle as Restricted Data in Foreign Dissemination, Section 144.b, Atomic Energy Act, 1954.

11.3.1.4 Elements <DocumentTitle>, <DocumentOriginator>, <OrganizationName>, <OrganizationAddress>, and <DocumentDate>

DOE manual 470.4-4A requires all documents to have a title or subject, originator, originator organization name and address, and date. If the information package is marked as a document, these elements contain the document title or subject, originator name and address, and document date.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
DocumentTitle	xsd:normalized-String	No	Document title. Element content is the title marked as required by DOE information marking manuals.
DocumentOriginator	xsd:normalized-String	No	Document originator. Element content is the originator such as the name of the document author.
OrganizationName	xsd:normalized-String	No	Originating organization name. Element content is the name.
OrganizationAddress	xsd:normalized-String	No	Originating organization address. Element content is the address.
DocumentDate	xsd:token	No	Document date. Element content is a date in any acceptable format.

The title or subject is placed in the <DocumentTitle> element and is marked as required by DOE manual 470.4-4A. The document originator is identified in <DocumentOriginator> and the originator's organization name and address are in <OrganizationName> and <OrganizationAddress> elements. Both the name and address are entered as text strings. The address text string should follow as closely as possible the format recommended for an address by the US Postal Service. The <DocumentDate> element can be used to specify a document date with the date in any appropriate date format. If a document date is not specified, the date in the <CreatedTimestamp> or <ModifiedTimestamp> element is the document date. The date in <DocumentDate> does not have to agree with the date in the <CreatedTimestamp> or <ModifiedTimestamp> elements. This date may preceded these dates such as when a document is created then stored in an information package.

11.3.1.5 Element <ClassifiedBy>

This element documents how the classification determination was made. If the package information was reviewed, it identifies the person or software application that performed the review. If the classification

level and category were set by default, it identifies the system whose highest classification level and category were used to set the information package classification level and category.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
ClassifiedBy	Empty	Yes	Identity of the person or software application that performed the review or the system whose highest classification level and category were used to set the classification level and category.

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
site	xsd:token	Req	Reviewer NSE two-letter site identifier
name	xsd:token	Req	Reviewer name or identifier of software application that performed the review. If attribute type has value default, it identifies the system whose highest classification level and category were used to set the information package classification level and category.
employeeId	xsd:token	Opt	Reviewer employee identifier such as employee number (if reviewed by a person)
title	xsd:token	Opt	Reviewer title (if reviewed by a person).
type	xsd:token	Req	How the review was performed. It must be one of these values: person — Review was performed by a person software — Review was performed by a software application default — Classification level and category was set to a default value unknown — How the review was performed is not known

Attribute site must contain the current two-letter NSE site identifier that identifies the site associated with the person or software application that made the classification determination. Attribute name must contain the name of the person that performed the review or the name and version of the software application that performed the review. If the classification level and category were set to a default value, it must contain the name of the system whose highest classification level and category were used to set the information package classification level and category. If the review was performed by a person, the site employee identifier associated with the person must be specified in the employeeId attribute and the title that person uses in classification markings must be specified in the title attribute.

11.3.1.6 Elements <DerivedFrom>, <DateReviewed> and <DeclassifyOn>

These elements contain the rest of the classifier markings required for classified documents by DOE manuals 470.4-4A and 475.1-1B. The source of the classification guidance used is identified in element <DerivedFrom> and the date the information was reviewed is in element <DateReviewed>. If all

classified package information is National Security Information, required declassification information is in element <DeclassifyOn>.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
DerivedFrom	xsd:normalized-String	No	Guidance used. Element content is guidance identification that conforms to the requirements of DOE manuals 470.4-4A and 475.1-1B.
DateReviewed	xsd:token	No	Information review date. Element content is a date in any acceptable date format.
DeclassifyOn	xsd:token	No	Declassification instructions. Element content contains instructions that meet the requirements of DOE manuals 470.4-4A and 475.1-1B.

Classification guidance identification in element DerivedFrom must conform to the requirements of DOE manuals 470.4-4A and 475.1-1B or their successors and to local guidance. Element DateReviewed can be used to record the date the information was reviewed. If this element is not specified, the date reviewed is the date in the <CreatedTimestamp> element. If all classified package information is National Security Information, element DeclassifyOn contains required declassification information.

These elements should be present only when the information package received a formal review from a person or a software application. They should not be present if the classification level and category were set to default values.

11.3.1.7 Element <AdditionalInformation>

Element <AdditionalInformation> is optional and can be used to provide more information about the classification determination and additional required information marking. The following are examples of information that can be stored in this element:

- Statement "Derivative Declassifier review required prior to declassification" required when all classified package information is National Security Information.
- Sources used to make the classification determination when multiple sources are used
- Classification level and category matrix when the information package contains information in multiple classification categories
- Additional information about the classification process used when a software application is used to make the classification determination

Element Tag Name	XSD Definition	Attr?	Element Description and Content
AdditionalInformation	xsd:string	No	Additional information. Element content is the additional information as general text.

Each independent statement should be in a separate <AdditionalInformation> element. If additional information is not required, this element should not be present. This element must not be

used to contain information when another <Classification> child element is a more appropriate container for that information.

11.3.2 Element <UnclassifiedControlled>

Unclassified controlled information (UCI) is unclassified information for which disclosure, loss, misuse, alteration, or destruction could adversely affect national security or governmental interests. Y-12 National Security Complex procedures in effect when this specification was written have instructions for identifying, marking, and handling twelve types of unclassified controlled information. Each of these twelve types has specific marking requirements that may differ slightly from other types. When an information package contains more than one type of unclassified controlled information, markings for every unclassified controlled information type present must be present in the information package.

Unclassified controlled information markings are contained in <UnclassifiedControlled> elements. Each <UnclassifiedControlled> element contains the markings for one type of unclassified controlled information. An information package that does not contain classified information must have one <UnclassifiedControlled> element that marks the unclassified controlled information type present or documents a determination that the information package contains only unclassified non-sensitive information. If the information package contains multiple types of unclassified controlled information, an <UnclassifiedControlled> element must be present for every type present.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
`UnclassifiedControlled`	Elements	No	Information marking for unclassified controlled information. Element content is One `<Type>` element Zero or more `<Caveat>` elements Zero or one `<DocumentTitle>` elements Zero or one `<DocumentOriginator>` elements Zero or one `<OrganizationName>` elements Zero or one `<OrganizationAddress>` elements Zero or one `<DocumentDate>` elements Zero or one `<Admonishment>` element Zero or one `<ClassifiedBy>` element Zero or one `<NameOrganization>` element Zero or one `<Reviewer>` element Zero or one `<ReviewingOfficial>` element Zero or one `<DerivedFrom>` element Zero or one `<Guidance>` element Zero or one `<GuidanceUsed>` element Zero or one `<DateReviewed>` element Zero or more `<AdditionalInformation>` elements

The information marking in element `<UnclassifiedControlled>` is the child elements listed above and described in alphabetical order below. Certain unclassified controlled information types require additional labeled information such as the identity of the person or entity that determined the unclassified controlled information type or the guidance used in that determination. The element tag names are designed to match as close as possible the labels used in the markings. In some cases elements with different tag names are used to represent basically the same information. This specification recommends that the information marking for an unclassified controlled information type use the element with the tag name that is the closest match to the label specified in the information type marking requirements. See Appendix C for more information on unclassified controlled information marking.

11.3.2.1 Element `<AdditionalInformation>`

The `<AdditionalInformation>` element is optional and provides additional information about the marking. It can be used, for example to contain the foreign government markings when the information package contains Confidential/Foreign Government Information–Modified Handling information. Each independent set of information should be in a separate `<AdditionalInformation>` element. If additional information is not required, this element should not be present. This element has the same definition as the `<AdditionalInformation>` element used in classification markings.

11.3.2.2 Element <Admonishment>

An admonishment statement is a statement that warns persons with access to the information about the consequences of releasing the information to unauthorized persons. The admonishment statement is placed in the element content. The statement will depend on the type of controlled information present. This element has the same definition as the <Admonishment> element used in classification markings.

11.3.2.3 Element <Caveat>

Element <Caveat> identifies special handling or dissemination requirements or assists in describing the type of information involved or who distributed or originated the information. The contents of this element will depend on the type of controlled information present. It should not be present if no caveats are required. This element has the same definition as the <Caveat> element used in classification markings.

11.3.2.4 Elements <ClassifiedBy> and <DerivedFrom>

Elements <ClassifiedBy> and <DerivedFrom> are used for information packages containing Confidential/Foreign Government Information–Modified Handling information. These elements have the same definitions as the <ClassifiedBy> and <DerivedFrom> elements used in classification.

11.3.2.5 Element <DateReviewed>

Element <DateReviewed> contains the date the information type determination was made. It has the same definition as the element used in the classification marking.

11.3.2.6 Element <Guidance> and <GuidanceUsed>

The guidance used to make a determination is entered in the element content of the <Guidance> and <GuidanceUsed> elements. The content of these elements is specified by the requirements of the information type.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
Guidance	xsd:normalized-String	No	Guidance used. Element content is guidance identification that conforms to the requirements of DOE manuals 470.4-4A and 475.1-1B.
GuidanceUsed	xsd:normalized-String	No	Guidance used. Element content is guidance identification that conforms to the requirements of DOE manuals 470.4-4A and 475.1-1B.

11.3.2.7 Elements <NameOrganization>, <ReviewingOfficial>, and <Reviewer>

These elements document how the information type determination was made. If the package information was reviewed, it identifies the person or software application that performed the review. If the unclassified controlled information type was set by default, it identifies the system on which the determination was based. The element used in the information package to identify the reviewer depends on the unclassified controlled information type. For example, <NameOrganization> is used for Official Use Only information and <ReviewingOfficial> is used for Unclassified Controlled Nuclear Information.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
NameOrganization	Empty	Yes	Identity of the person or software application that performed the review or the source of a default information type determination.
ReviewingOfficial	Empty	Yes	Identity of the person or software application that performed the review or the source of a default information type determination.
Reviewer	Empty	Yes	Identity of the person or software application that performed the review or the source of a default information type determination.

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
site	xsd:token	Req	Standard NSE two-letter site identifier
name	xsd:token	Req	Reviewer name or identifier of software application that performed the review. If attribute type has value default, it identifies the source of the default information type determination.
employeeId	xsd:token	Opt	Reviewer employee identifier such as employee number (if reviewed by a person)
title	xsd:token	Opt	Reviewer title (if reviewed by a person).
type	xsd:token	Req	How the review was performed. It must be one of these values: person — Review was performed by a person software — Review was performed by a software application default — Classification level and category was set to a default value unknown — How the review was performed is not known

The reviewer in all three elements is identified in the same set of element attributes. Attribute site contains the current two-letter NSE site identifier that identifies the site associated with the person or software application that performed the review. Attribute name contains the name of the person that performed the review or the name and version of the software application that performed the review. If the information type was set to a default value, it contains the name of the system on which the information type determination was based. If the review was performed by a person, the site employee identifier associated with the person must be specified in the employeeId attribute and the title that person uses in information type markings must be specified in the title attribute.

11.3.2.8 Element <Type>

The <Type> element contains the name of the unclassified controlled information type in its element content.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
Type	xsd:token	No	Unclassified controlled information type. Element content is the unclassified information type or the phrase Not Controlled if the information package does not contain unclassified controlled information .

The unclassified controlled information type is placed in the element content. If the information type is one of the controlled information types described in Y-12 procedures, the name of the controlled information type must be formatted as shown in this list:

```
Unclassified Controlled Nuclear Information
Export Controlled Information
Naval Nuclear Propulsion Information
Safeguards Information
Sensitive Nuclear Technology
Official Use Only
Applied Technology
Cooperative Research and Development Agreement
Confidential/Foreign Government Information-Modified Handling
Privacy Act information
Proprietary Information
Contractor Information
```

If the information package does not contain unclassified information, the following must be entered in the element content:

```
Not Controlled
```

The information type should be spelled and capitalized exactly as shown above. Only one blank must separate the words in the type name. If a new type of controlled information is defined, the name of the type should be formatted the same way.

11.4 ELEMENTS <AccessControl> AND <InfoAttribute>

Element <AccessControl> contains information attributes that can be used to determine whether a user or software application is allowed access to the package information. All information attributes significant to access determination except information already present in the <InformationMarking> element should be contained in this element.

This element is designed to provide information management systems with the information they need to determine whether these systems should allow or deny access to package information. Information

attributes are organized as name-value pairs with each name-value pair in a separate <InfoAttribute> element. An information management system is expected to get the access control information attributes it needs from the <AccessControl> element either when the information package is stored in the system or when an access request is received. The information management system is expected to use these attributes along with information attributes read from the <InformationMarking> element to determine whether an access request should be allowed or denied.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
AccessControl	Elements	No	Information attributes. Element content is Zero or more <InfoAttribute> elements

Element <InfoAttribute> contains an information attribute organized as a name-value pair. The information attribute name is in attribute name and its value is in element content. Any number of <InfoAttribute> elements can be included in the <AccessControl> element when the package is created and new elements can be added and existing elements modified or deleted as required over time.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
InfoAttribute	xsd:token	Yes	Information attribute. Element content is the information attribute value.

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
name	xsd:token	Req	Information attribute name.

11.5 ELEMENTS <SearchTerms> AND <SearchTerm>

These elements contain a set of search terms that can be used to locate the information package when it is stored in an information management system. The search terms are stored in <SearchTerm> elements each containing one search term in its element content and optional attributes that specify the search term name and search term units of measure. These <SearchTerm> elements are stored in the <SearchTerms> element content.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
SearchTerms	Elements	No	Search terms. Element content is Zero or more <SearchTerm> elements

Each <SearchTerm> element contains a single search term in its element content and optional attributes that specify the search term name and units of measure.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
SearchTerm	xsd:token	Yes	Search term. Element content is the search term.

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
name	xsd:token	Opt	Search term name
units	xsd:token	Opt	Search term units of measure

An information package can have as many <SearchTerm> elements as required to contain all of the terms that may be used to locate the information package.

11.6 ELEMENTS <References> AND <Reference>

Element <References> contains <Reference> elements each of which identifies another information package related in some way to the package with the <Reference> element. The <Reference> element has the same attributes as the <PackageIdentifier> element and the site, base identifier, revision, and instance of the related information package can be specified in them. The relationship is described by a phrase in the <Reference> element content.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
References	Elements	No	Related information package references. Element content is Zero or more <Reference> elements

Element Tag Name	XSD Definition	Attr?	Element Description and Content
Reference	xsd:normalized-String	Yes	Related information package reference. Element content is a phrase that describes the relationship.

The Reference element has the same attributes as the <PackageIdentifier> element with the same definitions. These attributes are defined in section 11.2.1 above.

A <Reference> element does not have to include all of the attribute values specified in the related information package <PackageIdentifier> element. For example, the related information package may be part of a series identified by versions. Each of these information packages has a specified site, base identifier, revision, and instance. The <Reference> element must specify the site and base identifier. If the relationship is to the series, it will not include the instance and may not include the revision.

11.7 ELEMENTS <History> AND <Event>

The <History> element documents events in the history of the information package or its information. Each event is described in an <Event> element that describes the event and specifies the source of the event, a timestamp that specifies when the event occurred, and whether package information was changed. <Event> elements are contained in the <History> element and are in chronological order.

The history can include events that occurred before the information package was created, while it was being created, and after it was created. Events that occurred before the information package was created and occur while it is being created are added to the history when the information package is created. Events that occur after the information package was created are added at the time they occur. In this way the <History> element can be used to record a history of the information package and the package information.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
History	Elements	No	Events in the history of the information package or package information. Element content is Zero or more <Event> elements

<Event> elements should be in chronological order in the <History>element content with the oldest immediately following the <History> element. An <Event> element should not be removed from the <History> element.

The <Event> element records information about a single event in the history of the information package or the package information. This event can be data creation, data revision, package creation, package revision, or any other event that impacts the information or information package. The attributes on the <Event> element describe the source of the event information. Event information includes the identity of the person or program that added the event, the time the event was added, and whether the event changed the information in the <PackageInfo> element.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
Event	xsd:string	Yes	Event in the history of the information package or package information. Element content is a description of the event.

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
site	xsd:token	Req	Standard NSE two-letter site identifier
name	xsd:token	Req	Source of the event. The source can be specified as the name of the person or the identifier of software application.
employeeId	xsd:token	Opt	Source employee identifier such as employee number (if the source is a person)
time	xsd:dateTime	Req	Date and time of the event in standard format.
packageInfoChanged	xsd:token	Req	Documents whether the package information was changed. It must be one of these values: yes — Packaged information was changed no — Packaged information was not changed unknown — Whether package information was changed is not known

If the source of the event is a program, attribute `name` contains the program name and version and attribute `employeeId` is absent. If the source of the event is a person, attribute `name` contains the person's name and attribute `employeeId` contains the person's employee number, badge number, or similar value. In both cases attribute `site` specifies the site and attribute `time` specifies the date and time of the event. The event description is entered in `<Event>` element content. This description can be standard text or any text desired by the event source.

11.8 ELEMENTS <Notes> AND <Note>

A `<Notes>` element allows users to add additional information to the information package about the information package or package information. Examples of appropriate notes include:

- Source of the information
- Quality of the information
- Reason the information package was modified

Each note is in a `<Note>` element that specifies the source of the note and a timestamp that specifies when the note was created. `<Note>` elements are contained in the `<Notes>` element and are in chronological order.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
`Notes`	Elements	No	Notes about the information package or package information. Element content is Zero or more `<Note>` elements

`<Note>` elements should be in chronological order with the oldest immediately following `<Notes>`. Each `<Note>` element contains a single note. A note should be restricted to one topic. Multiple topics should be in separate `<Note>` elements with one topic per element.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
`Note`	xsd:string	Yes	Note about the information package or package information. Element content is the text of the note.

Attributes of `<Note>` describe the source of the note. Source information includes the identity of the person or program that added the note and the time the note was added.

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
`site`	xsd:token	Req	Site identifier
`name`	xsd:token	Req	Person or program name
`employeeId`	xsd:token	Opt	Employee identifier (if known)
`time`	xsd:dateTime	Req	Date and time the information package was updated

If the source of the note is a program, attribute `name` contains the program name and version and attribute `employeeId` is absent. If the source of the note is a person, attribute `name` contains the person's name and attribute `employeeId` contains the person's employee number, badge number, or similar value. In both cases attribute `site` specifies the site and attribute `time` specifies the date and time the note was added.

11.9 ELEMENTS <PackageInfo>AND <PackageInfoRoot>

Element <PackageInfo> is the parent element for the elements at the top of the package information XML structure. It can also contain an XML Signature <Signature> element used to detect changes to and identify the source of the package information.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
PackageInfo	Elements	Yes	Parent element of the package information XML document root elements. Element content is One or more elements that conform to the <PackageInfoRoot> element definition Zero or one <Signature> element

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
id	xsd:ID	Opt	Identifies the <PackageInfo> element in an XML Signature

Attribute `id` is used only in an XML Signature to identify the <PackageInfo> element. This element is not needed if <PackageInfo> element content is not protected using an XML Signature.

Element <PackageInfoRoot> is used in this specification and in information packages defined using XML Schema definitions to define package information XML document root elements. Each package information XML document root element must have an element tag name that describes the information in it and a `version` attribute that specifies the package information XML document definition version. If element content is protected by an XML Signature, it must also have an `id` attribute that identifies the element in the XML Signature. If the information package document root and package metadata elements are in a namespace, every package information document root element must have an `xmlns` attribute that specifies a package information namespace for the elements below it in the package information XML document.

Element Tag Name	XSD Definition	Attr?	Element Description and Content
PackageInfoRoot		Yes	Package information XML document root element. Element content is Package information XML document elements Zero or one <Signature> element

The actual element tag name will be a name appropriate for the package information.

Attribute Name	XSD Definition	Reqd?	Attribute Description and Content
`version`	xsd:ID	Opt	Package information structure version
`id`	xsd:ID	Opt	Identifies the element in an XML Signature

A package info root element is defined in XML schema by defining it as a replacement for the `<PackageInfoRoot>` element in the information package XML document structure. The `<PackageInfoRoot>` element definition provides the element type required to define package information XML document root elements.

A `<PackageInfoRoot>` element cannot be defined using a document type definition (DTD) XML document definition. Each package information root element must be individually defined in the DTD and must have the attributes listed above for the `<PackageInfoRoot>` element.

ACRONYMS

AT	Applied Technology
C/FGI-MOD	Confidential/Foreign Government Information—Modified Handling
CRADA	Cooperative Research and Development Agreement
DOE	Department of Energy
DOE/NE	DOE Office of Nuclear Energy, Science, and Technology
DSS	Digital Signature Standard
DTD	Document Type Definition
ECI	Export Controlled Information
FIPS	Federal Information Processing Standard
FOIA	Freedom of Information Act
FRD	Formerly Restricted Data
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
NIST	National Institute of Standards and Technology
NSI	National Security Information
NNPI	Naval Nuclear Propulsion
NNPA	Nuclear Nonproliferation Act
NNSA	National Nuclear Security Administration
NSE	Nuclear Security Enterprise
OUO	Official Use Only
PRIDE	Product Realization Integrated Digital Enterprise
PCS	Product Characterization System
RD	Restricted Data
SAIC	Science Applications International Corporation
SHA	Secure Hash Algorithm
SI	Safeguards Information International System of Weights and Measures
SNT	Sensitive Nuclear Technology
UCI	Unclassified Controlled Information
UTC	Coordinated Universal Time
UCNI	Unclassified Controlled Nuclear Information
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema Definition

REFERENCES

[C14N10]
J. Boyer, ed. *Canonical XML Version 1.0.* W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/xml-c14n15 March 2001.

[C14N11]
Canonical XML Version 1.1. W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/xml-c14n11/2 May 2008.

[CMPC Marking]
CMPC Marking Resource: Examples of Acceptably Marked Classified or Controlled Matter. US Department of Energy, Washington DCSeptember 2009.

[DOE M 470.4-4A]
Information Security Manual. DOE manual 470.4-4AU.S. Department of Energy, January 16, 2009.

[DOE M 471.1-1]
Identification and Protection of Unclassified Controlled Nuclear Information Manual. DOE manual 471.1-1U.S. Department of Energy, October 23, 2001.

[DOE M 471.3-1]
Manual for Identifying and Protecting Official Use Only Information. DOE manual 471.3-1U.S. Department of Energy, April 9, 2003.

[DOE M 475.1-1B]
Manual for Identifying Classified Information. DOE manual 475.1-1BU.S. Department of Energy, August 28, 200.

[FIPS 180-3]
Secure Hash Standard (SHS). Federal Information Processing Standards Publication 180-3National Institute of Standards and Technology, October 2008.

[FIPS 186-3]
Digital Signature Standard (DSS). Federal Information Processing Standards Publication 186-3National Institute of Standards and Technology, June 2009.

[ISO 8601]
Data elements and interchange formats – Information interchange – Representation of dates and times. ISO 8601:2004International Organization for Standardization, 2004.

[NIST 44]
Specifications, Tolerances, and Other Technical Requirements for Weighing and Measuring Devices. NIST Handbook 44, 2009 editionNational Institute of Standards and Technology, 2009.

[OAIS1]
Reference Model for an Open Archival Information System (OAIS). CCSDS 650.0-B-1Consultative Committee for Space Data Systems (CCSDS), Washington, DC, USAJanuary 2002.

[OAIS2]
Space data and information transfer systems – Open archival information system – Reference model. ISO 14721:2003International Organization for Standardization, 2003.

[RFC 2141]
R. Moats. *URN Syntax.* The Internet Society, May 1997.

[RFC 3406]
L. Daigle, D. W. van Gulik, R. Iannella, P. Faltstrom. *Uniform Resource Names (URN) Namespace Definition Mechanisms.* The Internet Society, October 2002.

[RFC 3986]
Uniform Resource Identifier (URI): Generic Syntax . RFC 3986Internet Engineering Task Force (IETF), http://www.ietf.org/rfc/rfc3986.txtJanuary 2005.

[RFC 3987]
Internationalized Resource Identifiers (IRIs) . RFC 3987Internet Engineering Task Force (IETF), http://www.ietf.org/rfc/rfc3987.txtJanuary 2005.
[RFC 4648]
The Base16, Base32, and Base64 Data Encodings. RFC 4648Internet Engineering Task Force (IETF), http://www.ietf.org/rfc/rfc4648.txtOctober 2006.
[SI]
Le Système international d'unités—The International System of Units (SI). International Bureau of Weights and Measures (BIPM), Sèvres, France, http://www.bipm.org/utils/common/pdf/si_brochure_8_en.pdf2006.
[W3C Datetime]
Date and Time Formats. W3C Note, World Wide Web Consortium (W3C), http://www.w3.org/TR/NOTE-datetimeaccessed on August 27, 2009, http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231.
[XML 2008]
Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and Francois Yergeau, eds. *Extensible Markup Language (XML) 1.0 (Fifth Edition).* W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/xml/26 November 2008.
[XML DB]
Oracle Berkeley DB XML. Oracle Corporation, http://www.oracle.com/technology/products/berkeley-db/xml/index.htmlRetrieved on September 23, 2009.
[XMLDSIG 2002]
Eastlake, Donald, et. al., eds. *XML-Signature Syntax and Processing.* W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/February 2002.
[XMLDSIG 2008]
Eastlake, Donald, et. al., eds. *XML-Signature Syntax and Processing (Second Edition).* W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/ 10 June 2008.
[XML Namespaces]
Tim Bray, Dave Hollander, Andrew Layman, and Richard Tobin, eds. *Namespaces in XML 1.0 (Second Edition).* W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/REC-xml-names/16 August 2006.
[XML Schema]
Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn, eds. *XML Schema Part 1: Structures Second Edition.* W3C Recommendation, World Wide Web Consortium (W3C), http://www.w3.org/TR/xmlschema-1/28 October 2004.
[Y19-206]
Manual for Unclassified Controlled Information. Management Requirements, Y19-206BWXT Y-12, L.L.C., 28 September 2006.
[Y/IT-193]
Matthew Kelleher, Rick Shipp, and James David Mason. *Taxonomy-Based Search and Retrieval Implementation at Y-12.* Y/IT-193Y-12 National Security Complex, Oak Ridge, Tennessee11 September 2008.
[Y/IT-278]
Matthew Kelleher, Rick Shipp, and James David Mason. *Information Package Specification Version 1.0.* Y/IT-278Y-12 National Security Complex, Oak Ridge, Tennessee28 September 2009.

APPENDIX A

INFORMATION PACKAGE EXAMPLE

The signed information package shown below was created for the information package specification working example. It contains the part 8001 weight collected by A. B. Jones using scale S001 . The information package includes the following package metadata: package identifier, information marking, access control, search terms, and package history. The package information XML document root element is the `<ProductInfo>` element and its child elements contain the part weight and its context.

```
<?xml version="1.0" encoding="UTF-8"?>
<InfoPackage version="1.1"
  xmlns="urn:x-y12.doe.gov:InfoPackage:InfoPackage:1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PackageIdentification>
    <PackageIdentifier site="OR" identifier="D010-00-8001-WT" instance="1"/>
    <PackageDescription>
      Information package specification version 1.1
      Example product information package
    </PackageDescription>
    <CreatedTimestamp>2009-09-01T01:01:00.000-04:00</CreatedTimestamp>
  </PackageIdentifiers>
  <InformationMarking reviewed="yes">
    <Classification>
      <Level>Unclassified</Level>
    </Classification>
    <UnclassifiedControlled>
      <Type>Not Controlled</Type>
      <Reviewer site="OR" name="Matthew Kelleher" employeeId="000000"
        title="IT Software Engineer" type="person"/>
    </UnclassifiedControlled>
  </InformationMarking>
  <AccessControl>
    <InfoAttribute name="drawing">D010</InfoAttribute>
    <InfoAttribute name="material">Steel1</InfoAttribute>
  </AccessControl>
  <SearchTerms>
    <SearchTerm name="drawing">D010-00</SearchTerm>
    <SearchTerm name="material">Steel1</SearchTerm>
    <SearchTerm name="serial number">8001</SearchTerm>
    <SearchTerm name="information type">inspection</SearchTerm>
    <SearchTerm name="inspection type">weight</SearchTerm>
    <SearchTerm name="weight" units="kg">.991</SearchTerm>
  </SearchTerms>
  <References/>
  <History>
    <Event site="OR" name="A. B. Jones" employeeId="E101"
      time="2009-09-01T01:01:00.000-04:00"
```

```
    packageInfoChanged="yes”>
    Package created.
    </Event>
</History>
<Notes/>
<PackageInfo>
  <ProductInfo version="1.0" id="SignedContents"
    xmlns="urn:x-y12.doe.gov:InfoPackage:ProductInfo:1.0">
    <Contents>
      <InformationType>weight measurement</InformationType>
      <Description>Part weight</Description>
    </Contents>
    <Product>
      <DrawingNumber>D010-00</DrawingNumber>
      <SerialNumber>8001</SerialNumber>
    </Product>
    <Process>
      <Facility>F001</Facility>
      <Route>D010Route</Route>
      <Operation>WT01</Operation>
    </Process>
    <Data>
      <DataValue>
        <Name>weight</Name>
        <Value units="g">991.</Value>
      </DataValue>
      <DataValue>
        <Name>machine identifier</Name>
        <Value>S001</Value>
      </DataValue>
    </Data>
    <Operators>
      <Operator site="OR" name="A. B. Jones" employeeId="E101"
        time="2009-09-01T01:01:00.000-04:00"/>
    </Operators>
  </ProductInfo>
  <Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
      <Reference
        URI="#SignedContents">
        <Transforms>
          <Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        </Transforms>
        <DigestMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
```

```
            <DigestValue>Jo4tFDqWogvDAddU/IUEaSDp4jg=</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>
          RBjMfRUd4/gp5iUtAws/369mmLUQfAgFTCvUoB3Nz+NkgcCXbggwMQ==
        </SignatureValue>
        <KeyInfo>
          <X509Data>
            <X509SubjectName>
              CN=infopackage.y12.gov,OU=Information Technology,O=Y-12 Compex,L=Oak Ridge,ST=Tenness
            </X509SubjectName>
            <X509Certificate>
              MIIDTzCCAw2gAwIBAgIESrErxjALBgcqhkjOOAQDBQAwgYoxCzAJBgNVBAYTAlVTMRIwEAYDVQQI
              EwlUZW5uZXNzZWUxEjAQBgNVBAcTCU9hayBSaWRnZTEUMBIGA1UEChMLWS0xMiBDb21wZXgxHzAd
              BgNVBAsTFkluZm9ybWF0aW9uIFRlY2hub2xvZ3kxHDAaBgNVBAMTE2luZm9wYWNrYWdlLnkxMi5n
              b3YwHhcNMDkwOTE2MTgxNzQyWhcNMTEwOTA2MTgxNzQyWjCBijELMAkGA1UEBhMCVVMxEjAQBgNV
              BAgTCVRlbm5lc3NlZTESMBAGA1UEBxMJT2FrIFJpZGdlMRQwEgYDVQQKEwtZLTEyIENvbXBleDEf
              MB0GA1UECxMWSW5mb3JtYXRpb24gVGVjaG5vbG9neTEcMBoGA1UEAxMTaW5mb3BhY2thZ2UueTEy
              LmdvdjCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQD9f1OBHXUSKVLfSpwu7OTn9hG3UjzvRADDHj+A
              tlEmaUVdQCJR+1k9jVj6v8X1ujD2y5tVbNeBO4AdNG/yZmC3a5lQpaSfn+gEexAiwk+7qdf+t8Yb
              +DtX58aophUPBPuD9tPFHsMCNVQTWhaRMvZ1864rYdcq7/IiAxmd0UgBxwIVAJdgUI8VIwvMspK5
              gqLrhAvwWBz1AoGBAPfhoIXWmz3ey7yrXDa4V7l5lK+7+jrqgvlXTAs9B4JnUVlXjrrUWU/mcQcQ
              gYC0SRZxI+hMKBYTt88JMozIpuE8FnqLVHyNKOCjrh4rs6Z1kW6jfwv6ITVi8ftiegEkO8yk8b6o
              UZCJqIPf4VrlnwaSi2ZegHtVJWQBTDv+z0kqA4GEAAKBgCm9ncRDbugOvR9LGRMXnFCu2u5BAAJJ
              f3iLB108E3B6ymsa1zcwq7uemNvgZnzFNm4UfgcMRuhnUenyJIwCoSu8a91N0vQXSuewS7t0Z/7M
              Ge9vwH494bfvAJ8MBIe5qlMLKelTYr5fnGJPjsWPQ3yFVLqo/SHqQNqs+qX8wBc2MAsGByqGSM44
              BAMFAAMvADAsAhQbnAxxEbo67W3tl+Kq/qzcM3yfEQIUXP7XGNZ9D6CxnjDntlBsMfkt4/Q=
            </X509Certificate>
          </X509Data>
        </KeyInfo>
      </Signature>
    </PackageInfo>
  </InfoPackage>
```

APPENDIX B

INFORMATION PACKAGE XML DOCUMENT DEFINITIONS

A complete information package XML document is composed of the information package document root and package metadata described in this specification and one or more package information XML documents. The XML Schema definition (XSD) and document type definition (DTD) of the information package document root and package metadata are described in this appendix. Package information XML document definitions are outside the scope of this specification.

The XML Schema and document type definitions in this appendix define these information package elements:

```
<InfoPackage>
<PackageIdentification>
<InformationMarking>
<AccessControl>
<SearchTerms>
<References>
<History>
<Notes>
<PackageInfo>
```

Element `<InfoPackage>` is the information package document root element, `<PackageInfo>` is the parent element of the package information, and the other elements are package metadata elements.

The XML Schema and DTD elements shown in this appendix were formatted to make them easy to understand and use. A production implementation may rearrange these elements to make them easier to maintain.

B.1 XML SCHEMA DEFINITION

The information package XML Schema definition starts with the `<schema>` element that defines the XML Schema document. Its element attributes define the recommended `ip` namespace used to define information package elements and the recommended `ds` namespace used to define XML Signature elements. It sets the target namespace to the information package namespace:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ip="urn:x-y12.doe.gov:InfoPackage:InfoPackage:1.1"
  targetNamespace="urn:x-y12.doe.gov:InfoPackage:InfoPackage:1.1"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  elementFormDefault="qualified">
```

The XML Schema that define XML Signature elements is imported using this statement:

```
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#"/>
```

This namespace is required because the XML Signature `<Signature>` element is an optional child element of `<PackageInfo>`.

The information package XML Schema definition, package information XSD Schema definition, and XML Signature XML Schema definition can be placed in a separate files. The information package and XML Schema files can be imported into the package information XML file to provide it with the definitions of the `<PackageInfoRoot>` and `<Signature>` elements.

XML Schema definitions used to define information package element and attribute contents use the content models defined in XML Schema. These content models were developed to define XML documents that contain structured information and are much more precise than the CDATA and PCDATA content models available in document type definitions.

B.2 DOCUMENT TYPE DEFINITION

The document type definition for the information package does not have any headers or other required supplemental material. The element and attribute definitions shown in this appendix define the information package.

Document type definitions do not have a natural way of adding the contents of other files to a DTD file. If a document type definition is used to specify an information package, the information package, package information, and XML Signature XML definitions may have to be combined into a single file.

XML documents were originally designed to contain general text information. The document type definition for XML was created at the same time and was designed to define documents that contain general text information. The CDATA and PCDATA content models available in DTDs are appropriate for these uses.

B.3 ATTRIBUTE GROUPS AND ENTITIES

Attribute groups and entities are defined to provide common definitions for attributes used in multiple elements. Three sets of attributes are defined for information packages:

Package identification attributes — Defines attributes that store package identification information
Actor attributes — Defines attributes that identify persons and software applications
Action attributes — Defines attributes that record actions taken by an actors

B.3.1 Package Identification Attributes

The attributes used to identify information packages are defined in an XML Schema attribute group and in a DTD entity. The attributes defined in the attribute group and entity are:

`site` — NSE two-letter site identifier

`identifier` — Information package base identifier
`revision` — Information package revision
`instance` — Information package instance

The XML Schema element structure that defines the package identification attribute group is:

```
<xsd:attributeGroup name="IdentifierAttrGroup">
  <xsd:attribute name="site" type="xsd:token" use="required"/>
  <xsd:attribute name="identifier" type="xsd:token" use="required"/>
  <xsd:attribute name="revision" type="xsd:token" use="optional"/>
  <xsd:attribute name="instance" type="xsd:token" use="optional"/>
</xsd:attributeGroup>
```

The DTD definition for the package identification attribute entity is

```
<!ENTITY % IdentifierAttrEntity
  '  site CDATA #REQUIRED
     identifier CDATA #REQUIRED
     revision CDATA #IMPLIED
     instance CDATA #IMPLIED
 ' >
```

B.3.2 Actor Attributes

The attributes used to identify an actor are defined in an XML Schema attribute group and in a DTD entity. An actor is a person or a software application. The attributes defined in the attribute group and entity are:

`site` — NSE two-letter site identifier
`name` — Information package base identifier
`employeeId` — Information package revision

A person is identified by a `site`, `name`, and `employeeId` identifier. A software application is identified by a `site` and a name and version stored together in the `name` attribute.

The XML Schema element structure that defines the actor attribute group is:

```
<xsd:attributeGroup name="ActorAttrGroup">
  <xsd:attribute name="site" type="xsd:token" use="required"/>
  <xsd:attribute name="name" type="xsd:token" use="required"/>
  <xsd:attribute name="employeeId" type="xsd:token" use="optional"/>
</xsd:attributeGroup>
```

The DTD definition for the actor attribute entity is

```
<!ENTITY % ActorAttrEntity
  '  site CDATA #REQUIRED
     name CDATA #REQUIRED
     employeeId CDATA #IMPLIED
  '>
```

B.3.3 Action Attributes

An actor as defined above performs an action that affects the information package. Such an action may be to create or modify the information package or to add a note about the information package or package information. Action attributes record the identity of the actor and the date and time the action was performed:

Actor attribute group — Person or software application that performed the action
`time` — Date and time the action occurred recorded as a timestamp

The XML Schema element structure that defines the action attribute group is:

```
<xsd:attributeGroup name="ActionAttrGroup">
  <xsd:attributeGroup ref="ip:ActorAttrGroup"/>
  <xsd:attribute name="time" type="xsd:dateTime" use="required"/>
</xsd:attributeGroup>
```

This definition assumes the timestamp used to record the date and time has the format recommended in section 8. If it has another format, the XML Schema simple type used to define attribute `time` should be `xsd:token`

The DTD definition for the action attribute entity is

```
<!ENTITY % ActionAttrEntity
  '  %ActorAttrEntity;
     time CDATA #REQUIRED
  '>
```

B.4 ELEMENT `<InfoPackage>`

This element is the document root of the information package. Its child elements contain the information package package metadata and its `<PackageInfo>` element contains the package information XML document root elements.

The XML Schema element structure that defines this element is

```
<xsd:element name="InfoPackage">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element ref="ip:PackageIdentifiers" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ip:InformationMarking" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ip:AccessControl" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ip:SearchTerms" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:References" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:History" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:Notes" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:PackageInfo" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="version" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:token">
          <xsd:enumeration value="1.1"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

The definition of attribute version restricts its values to the version number of this specification. New version numbers can be added to the enumeration if necessary.

The DTD definition for this element is

```
<!ELEMENT InfoPackage (PackageIdentifiers , InformationMarking , AccessControl
    , SearchTerms? , References? , History? , Notes? , PackageInfo) >
<!ATTLIST InfoPackage
  version CDATA #REQUIRED >
```

B.5 ELEMENT `<PackageIdentification>`

This child element of `<InfoPackage>` contains package identification information in its child elements.

The XML Schema element structure that defines these elements is

```
<xsd:element name="PackageIdentification">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ip:PackageIdentifier" minOccurs="1" maxOccurs="1">
      <xsd:element ref="ip:PredecessorIdentifier" minOccurs="0" maxOccurs="1">
      <xsd:element ref="ip:SuccessorIdentifier" minOccurs="0" maxOccurs="1">
      <xsd:element ref="ip:AlternateIdentifier" minOccurs="0"  maxOccurs="unbounded">
      <xsd:element ref="ip:PackageDescription" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:PackageStatus" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:CreatedTimestamp" minOccurs="0"  maxOccurs="1"/>
      <xsd:element ref="ip:ModifiedTimestamp" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT PackageIdentification (PackageIdentifier , PredecessorIdentifier?
    , SuccessorIdentifier* , AlternateIdentifier* , PackageDescription?
    , PackageStatus? , CreatedTimestamp? , ModifiedTimestamp?) >
```

B.5.1 Elements <PackageIdentifier>, <PredecessorIdentifier>, and <SuccessorIdentifier>

These <PackageIdentification> child elements contain the package identification and the package identification of a predecessor information packages and of successor information packages.

The XML Schema element structure that defines these elements is

```
<xsd:element name="PackageIdentifier" minOccurs="1" maxOccurs="1">
  <xsd:complexType>
    <xsd:attributeGroup ref="ip:IdentifierAttrGroup"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="PredecessorIdentifier" minOccurs="0" maxOccurs="1">
  <xsd:complexType>
    <xsd:attributeGroup ref="ip:IdentifierAttrGroup"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="SuccessorIdentifier" minOccurs="0" maxOccurs="1">
  <xsd:complexType>
    <xsd:attributeGroup ref="ip:IdentifierAttrGroup"/>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT PackageIdentifier EMPTY >
<!ATTLIST PackageIdentifier
  %IdentifierAttrEntity; >
<!ELEMENT PredecessorIdentifier EMPTY >
<!ATTLIST PredecessorIdentifier
  %IdentifierAttrEntity; >
<!ELEMENT SuccessorIdentifier EMPTY >
<!ATTLIST SuccessorIdentifier
  %IdentifierAttrEntity; >
```

Attribute group IdentifierAttrGroup and attribute entity IdentifierAttrEntity define the site, identifier, revision, and instance attributes used to identify information packages.

B.5.2 Element <AlternateIdentifier>

This <PackageIdentification> child element contains an alternate identifier for the information package or package information.

The XML Schema element structure that defines these elements is

```
<xsd:element name="AlternateIdentifier" minOccurs="0"  maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:token">
        <xsd:attribute name="name" type="xsd:token" use="required"/>
        <xsd:attribute name="usedFor" use="required">
          <xsd:simpleType>
            <xsd:restriction base="xsd:token">
              <xsd:enumeration value="package"/>
              <xsd:enumeration value="information"/>
              <xsd:enumeration value="other"/>
              <xsd:enumeration value="unknown"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT AlternateIdentifier (#PCDATA) >
<!ATTLIST AlternateIdentifier
  name CDATA #REQUIRED
  usedFor (package|information|other|unknown) #REQUIRED >
```

B.5.3 Elements <PackageDescription> and <PackageStatus>

These <PackageIdentification> child elements contain the package description and package status in their element content.

The XML Schema element structure that defines these elements is

```
<xsd:element name="PackageDescription" type="xsd:string" minOccurs="0" maxOccurs="1"/>
<xsd:element name="PackageStatus" type="xsd:token" minOccurs="0" maxOccurs="1"/>
```

The corresponding DTD definition is

```
<!ELEMENT PackageDescription (#PCDATA) >
<!ELEMENT PackageStatus (#PCDATA) >
```

B.5.4 Elements <CreatedTimestamp> and <ModifiedTimestamp>

These <PackageIdentification>child elements contain the date and time the information package was created and the date and time the information package was last modified. Dates and times are stored as timestamps in element content.

The XML Schema element structure that defines these elements is

```
<xsd:element name="CreatedTimestamp" type="xsd:dateTime" minOccurs="0"  maxOccurs="1"/>
<xsd:element name="ModifiedTimestamp" type="xsd:dateTime" minOccurs="0" maxOccurs="1"/>
```

This structure assumes that the timestamps are stored in the form recommended in section 8. If another form is used, the XML Schema simple type that defines element content should be xsd:token

The corresponding DTD definition is

```
<!ELEMENT CreatedTimestamp (#PCDATA) >
<!ELEMENT ModifiedTimestamp (#PCDATA) >
```

B.6 ELEMENT <InformationMarking>

This element is a child element of <InfoPackage> and contains the information marking required to protect the information in the information package. Information marking for classified information is in child element <Classification>and for unclassified controlled information in <UnclassifiedControlled>.

The XML Schema element structure that defines this element is

```
<xsd:element name="InformationMarking">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ip:Classification" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ip:UnclassifiedControlled" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:AdditionalInformation" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="reviewed" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:token">
          <xsd:enumeration value="yes"/>
          <xsd:enumeration value="no"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT InformationMarking (Classification , UnclassifiedControlled?
```

```
    , AdditionalInformation*) >
<!ATTLIST InformationMarking
  reviewed (yes|no) #REQUIRED >
```

B.6.1 Element <Classification>

This child element of <InformationMarking>contains the information marking required for classified information. The information marking is in child elements whose tag names identify the marking contained in them.

The XML Schema element structure that defines this element is

```
<xsd:element name="Classification">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ip:Level" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ip:Category" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:Caveat" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="ip:SpecialControlMarking" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="ip:DocumentTitle" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DocumentOriginator" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:OrganizationName" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:OrganizationAddress" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DocumentDate" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:Admonishment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:ClassifiedBy" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DerivedFrom" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DeclassifyOn" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DateReviewed" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:AdditionalInformation" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT Classification (Level , Category? , Caveat* , SpecialControlMarking*
    , DocumentTitle? , DocumentOriginator? , OrganizationName?
    , OrganizationAddress? , DocumentDate? , Admonishment? , ClassifiedBy?
    , DerivedFrom? , DeclassifyOn? , DateReviewed? , AdditionalInformation*) >
```

B.6.2 Element <UnclassifiedControlled>

This child element of <InformationMarking> contains the information marking required for unclassified controlled information. The information marking is in child elements whose tag names identify the marking contained in them.

The XML Schema element structure that defines this element is

```
<xsd:element name="UnclassifiedControlled">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ip:Type" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ip:Caveat" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="ip:DocumentTitle" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DocumentOriginator" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:OrganizationName" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:OrganizationAddress" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DocumentDate" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:Admonishment" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:ClassifiedBy" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:NameOrganization" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:Reviewer" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:ReviewingOfficial" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DerivedFrom" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:Guidance" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:GuidanceUsed" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:DateReviewed" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ip:AdditionalInformation" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT UnclassifiedControlled (Type , Caveat* , DocumentTitle?
    , DocumentOriginator? , OrganizationName? , OrganizationAddress?
    , DocumentDate? , Admonishment? , ClassifiedBy? , NameOrganization?
    , Reviewer? , ReviewingOfficial? , DerivedFrom? , Guidance?
    , GuidanceUsed? , DateReviewed? , AdditionalInformation*) >
```

B.6.3 Elements <Level>, <Category>, and <Type>

These elements contain the classification level and category and the unclassified information type. Elements <Level> and <Category> are child elements of <Classification> and element <Type> is a child element of <UnclassifiedControlled>. The classification level, classification category, and unclassified controlled information type is in element content.

The XML Schema element structure that defines these elements is

```
<xsd:element name="Level">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="Top Secret"/>
      <xsd:enumeration value="Secret"/>
      <xsd:enumeration value="Confidential"/>
      <xsd:enumeration value="Unclassified"/>
    </xsd:restriction>
```

```
    </xsd:simpleType>
  </xsd:element>
    <xsd:element name="Category">
    <xsd:simpleType>
      <xsd:restriction base="xsd:token">
        <xsd:enumeration value="Restricted Data"/>
        <xsd:enumeration value="Formerly Restricted Data"/>
        <xsd:enumeration value="National Security Information"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="Type" type="xsd:token"/>
```

The acceptable values for <Level> and <Category> element content are limited to the standard classification levels and categories. The acceptable values for <Type> are not limited in this way because unclassified controlled information types are likely to change over time.

The corresponding DTD definition is

```
<!ELEMENT Level (#PCDATA) >
<!ELEMENT Category (#PCDATA) >
<!ELEMENT Type (#PCDATA) >
```

B.6.4 Elements <Caveat>, <SpecialControlMarking>, and<Admonishment>

These elements are used to store caveats, special control markings, and admonishments required to mark the information package. Elements <Caveat> and<Admonishment> are child elements of both <Classification> and <UnclassifiedControlled> elements. Element <SpecialControlMarking> is a child element only of element <Classification>.

The XML Schema element structure that defines these elements is

```
<xsd:element name="Caveat" type="xsd:token"/>
<xsd:element name="SpecialControlMarking" type="xsd:token"/>
<xsd:element name="Admonishment" type="xsd:string"/>
```

The corresponding DTD definition is

```
<!ELEMENT Caveat (#PCDATA) >
<!ELEMENT SpecialControlMarking (#PCDATA) >
<!ELEMENT Admonishment (#PCDATA) >
```

B.6.5 Elements <DocumentTitle>, <DocumentOriginator>, <OrganizationName>, <OrganizationAddress>, and <DocumentDate>

These elements are used to store document information when the information package is marked as a document. All five elements are child elements of both <Classification> and <UnclassifiedControlled> elements.

The XML Schema element structure that defines these elements is

```
<xsd:element name="DocumentTitle" type="xsd:normalizedString"/>
<xsd:element name="DocumentOriginator" type="xsd:normalizedString"/>
<xsd:element name="OrganizationName" type="xsd:normalizedString"/>
<xsd:element name="OrganizationAddress" type="xsd:normalizedString"/>
<xsd:element name="DocumentDate" type="xsd:token"/>
```

The corresponding DTD definition is

```
<!ELEMENT DocumentTitle (#PCDATA) >
<!ELEMENT DocumentOriginator (#PCDATA) >
<!ELEMENT OrganizationName (#PCDATA) >
<!ELEMENT OrganizationAddress (#PCDATA) >
<!ELEMENT DocumentDate (#PCDATA) >
```

B.6.6 Elements <ClassifiedBy>, <NameOrganization>, <Reviewer>, and <ReviewingOfficial>

These elements are used to identify the person or software application that reviewed the information package. Element <ClassifiedBy> is a child element of both <Classification> and <UnclassifiedControlled> and the other three are child elements of <UnclassifiedControlled>.

The XML Schema element structure that defines these elements is

```
<xsd:complexType name="ReviewerElementType">
  <xsd:attributeGroup ref="ip:ActorAttrGroup"/>
  <xsd:attribute name="title" type="xsd:token" use="optional"/>
  <xsd:attribute name="type" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:token">
        <xsd:enumeration value="person"/>
        <xsd:enumeration value="software"/>
        <xsd:enumeration value="default"/>
        <xsd:enumeration value="unknown"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>
<xsd:element name="ClassifiedBy" type="ip:ReviewerElementType"/>
<xsd:element name="NameOrganization" type="ip:ReviewerElementType"/>
<xsd:element name="Reviewer" type="ip:ReviewerElementType"/>
```

```
  <xsd:element name="ReviewingOfficial" type="ip:ReviewerElementType"/>
```

The corresponding DTD definition is

```
<!ENTITY % reviewerAttrEntity
  '   %ActorAttrEntity;
      title CDATA #IMPLIED
      type (person|software|unknown) #REQUIRED
 ' >
<!ELEMENT ClassifiedBy EMPTY >
<!ATTLIST ClassifiedBy
  %reviewerAttrEntity; >
<!ELEMENT NameOrganization (#PCDATA) >
<!ATTLIST NameOrganization
  %reviewerAttrEntity; >
<!ELEMENT Reviewer (#PCDATA) >
<!ATTLIST Reviewer
  %reviewerAttrEntity; >
<!ELEMENT ReviewingOfficial (#PCDATA) >
<!ATTLIST ReviewingOfficial
  %reviewerAttrEntity; >
```

The identity of the reviewer is recorded in element attributes `site`, `name`, `employeeId`, `title` and `type`. These attributes are defined as an attribute group (XSD) or attribute entity (DTD) to ensure they have common definitions.

B.6.7 Elements <DerivedFrom>, <Guidance>, and <GuidanceUsed>

These elements contain the guidance used determine whether the information package contains classified or unclassified controlled information. Element `<DerivedFrom>` is a child element of both `<Classification>` and `<UnclassifiedControlled>` and the other two are child elements of `<UnclassifiedControlled>`.

The XML Schema element structure that defines these elements is

```
<xsd:element name="DerivedFrom" type="xsd:normalizedString"/>
<xsd:element name="Guidance" type="xsd:normalizedString"/>
<xsd:element name="GuidanceUsed" type="xsd:normalizedString"/>
```

The corresponding DTD definition is

```
<!ELEMENT DerivedFrom (#PCDATA) >
<!ELEMENT Guidance (#PCDATA) >
<!ELEMENT GuidanceUsed (#PCDATA) >
```

B.6.8 Elements <DateReviewed> and <DeclassifyOn>

These two elements provide dates required by the information marking. Element <DateReviewed> contains the date the information package was reviewed. If the information package contains National Security Information, element <DeclassifyOn> contains required declassification date or exemption. Element <DateReviewed> is a child element of both <Classification> and <UnclassifiedControlled> and <DeclassifyOn> is a child element of <Classification>.

The XML Schema element structure that defines these elements is

```
<xsd:element name="DateReviewed" type="xsd:token"/>
<xsd:element name="DeclassifyOn" type="xsd:token"/>
```

The corresponding DTD definition is

```
<!ELEMENT DateReviewed (#PCDATA) >
<!ELEMENT DeclassifyOn (#PCDATA) >
```

B.6.9 Element <AdditionalInformation>

This element is used to store additional information relevant to the classification or unclassified controlled information type determination process. It is a child element of <InformationMarking>, <Classification>, and <UnclassifiedControlled>.

The XML Schema element structure that defines this element is

```
<xsd:element name="AdditionalInformation" type="xsd:string"/>
```

The corresponding DTD definition is

```
<!ELEMENT AdditionalInformation (#PCDATA) >
```

B.7 ELEMENTS <AccessControl> AND <InfoAttribute>

These elements contain the information required to determine whether a user or application is granted or denied access to the package information. Element <AccessControl> is a child element of <InfoPackage> and contains <InfoAttribute> elements that store the access control information attributes.

The XML Schema element structure that defines these elements is

```
<xsd:element name="AccessControl">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="InfoAttribute" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:simpleContent>
```

```
            <xsd:extension base="xsd:token">
              <xsd:attribute name="name" type="xsd:token" use="required"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT AccessControl (InfoAttribute*) >
<!ELEMENT InfoAttribute (#PCDATA) >
<!ATTLIST InfoAttribute
    name CDATA #REQUIRED >
```

B.8 ELEMENTS <SearchTerms> AND <SearchTerm>

These elements that contain search terms that can be used to locate the information package. Element <SearchTerms> is a child element of <InfoPackage> and contains <SearchTerm> elements that store search terms.

The XML Schema element structure that defines these elements is

```
<xsd:element name="SearchTerms">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="SearchTerm" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:token">
              <xsd:attribute name="name" type="xsd:token" use="optional"/>
              <xsd:attribute name="units" type="xsd:token" use="optional"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT SearchTerms (SearchTerm*) >
<!ELEMENT SearchTerm (#PCDATA) >
<!ATTLIST SearchTerm
  name CDATA #IMPLIED
```

```
    units CDATA #IMPLIED >
```

B.9 ELEMENTS <References> AND <Reference>

These elements contain references to other information packages. Element <References> is a child element of <InfoPackage> and contains <Reference> elements that store the references.

The XML Schema element structure that defines these elements is

```
<xsd:element name="References">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Reference" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:normalizedString">
              <xsd:attributeGroup ref="ip:IdentifierAttrGroup"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT References (Reference*) >
<!ELEMENT Reference (#PCDATA) >
<!ATTLIST Reference
  %IdentifierAttrEntity; >
```

Attribute group IdentifierAttrGroup and attribute entity IdentifierAttrEntity define the site, identifier, revision, and instance attributes used to identify information packages.

B.10 ELEMENTS <History> AND <Event>

These elements contain the history of the information package and the package information. Element <History> is a child element of <InfoPackage> and contains <Event> elements that represent events in the history of the information package and package information.

The XML Schema element structure that defines these elements is

```
<xsd:element name="History">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Event" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
```

```
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                  <xsd:attributeGroup ref="ip:ActionAttrGroup"/>
                  <xsd:attribute name="packageInfoChanged" use="required">
                    <xsd:simpleType>
                    <xsd:restriction base="xsd:token">
                      <xsd:enumeration value="yes"/>
                      <xsd:enumeration value="no"/>
                      <xsd:enumeration value="unknown"/>
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:attribute>
              </xsd:extension>
            </xsd:simpleContent>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT History (Event*) >
<!ELEMENT Event (#PCDATA) >
<!ATTLIST Event
  %ActionAttrEntity;
  packageInfoChanged (yes|no|unknown) #REQUIRED >
```

Attribute group `ActionAttrGroup` and attribute entity `ActionAttrEntity` define the `site`, `name`, `employeeId`, and `time` attributes.

B.11 ELEMENTS <Notes> AND <Note>

These elements contain notes about the information package or the package information. Element `<Notes>` is a child element of `<InfoPackage>` and contains `<Note>` elements that contain notes about the information package and package information. The source of the note and the date and time it was made are recorded in `<Note>` element attributes `site`, `name`, `employeeId`, and `time`. Note text is in element content.

The XML Schema element structure that defines these elements is

```
<xsd:element name="Notes">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Note" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
```

```
              <xsd:attributeGroup ref="ip:ActionAttrGroup"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition is

```
<!ELEMENT Notes (Note*) >
<!ELEMENT Note (#PCDATA) >
<!ATTLIST Note
  %ActionAttrEntity; >
```

Attribute group `ActionAttrGroup` and attribute entity `ActionAttrEntity` define the `site`, `name`, `employeeId`, and `time` attributes.

B.12 ELEMENT <PackageInfo>

This element contains the package information. Its child elements are the package information XML structure document root elements. This element can also contain an XML Signature `<Signature>` element that signs the `<PackageInfo>` element or some or all of its child elements.

The XML Schema element structure that defines this element is:

```
<xsd:element name="PackageInfo">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ip:PackageInfoRoot" minOccurs="1" maxOccurs="unbounded"/>
      <xsd:element ref="ds:Signature" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" use="optional"/>
  </xsd:complexType>
</xsd:element>
```

The corresponding DTD definition for this element is:

```
<!ELEMENT PackageInfo (PackageInfoRoot+, ds:Signature?)>
<!ATTLIST PackageInfo
  id ID #IMPLIED >
```

B.13 PACKAGE INFORMATION XML DOCUMENTS

Package information is stored in XML documents that have a single document root element that is a child element of the `<PackageInfo>` element. Element `<PackageInfo>` can contain one or more

package information XML document root elements. Package information XML documents can also have XML Signature `<Signature>` elements along with the elements required to store the package information.

The information package XML Schema definition includes the definition of a `<PackageInfoRoot>` element. This element defines the required `version` and optional `id` attributes. It is designed to be replaced by a package information root element defined in a separate XML Schema document.

The `<PackageInfoRoot>` element has this XML Schema definition:

```
<xsd:element name="PackageInfoRoot" type="ip:PackageInfoRootType"/>
<xsd:complexType name="PackageInfoRootType">
  <xsd:attribute name="version" type="xsd:token" use="required"/>
  <xsd:attribute name="id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

An equivalent element definition does not exist in DTD definitions. A DTD must directly specify the package information XML document root elements in its definition of the `<PackageInfo>` element. The package information XML document root elements must include the required `version` and optional `id` attribute definitions as shown in this definition:

```
<!ELEMENT PackageInfoRoot ANY>
<!ATTLIST PackageInfoRoot
  version CDATA #REQUIRED
  id ID #IMPLIED >
>
```

APPENDIX C

UNCLASSIFIED CONTROLLED INFORMATION MARKING REQUIREMENTS

Section 5 specifies the marking requirements for Unclassified Controlled Nuclear Information (UCNI) and for Official Use Only (OUO) information and defines the elements used to mark other types of unclassified controlled information. This section specifies the marking requirements for other types of unclassified controlled information used at the Y-12 National Security Complex. The information in this section was obtained from Y-12 manual Y19-206 *Manual for Unclassified Controlled Information* [Y19-206].

In every case an admonitory marking must be added to the document to inform users that the specified type of information is present in the information package This admonitory marking has text in capital letters and may have text in sentence case. This capitalization must be used in the `<Admonishment>` element content.

Additional elements must be added in certain cases to identify the person or entity that made the unclassified controlled information type determination. These elements are specified in the section that requires them.

C.1 EXPORT CONTROLLED INFORMATION

Export Controlled Information is certain scientific and technical information products containing technical data as defined in and controlled by the International Traffic in Arms Regulations, Export Administration Regulations, Nuclear Nonproliferation Act of 1978 (NNPA), and the Atomic Energy Act of 1954, as amended.

If the information package contains Export Controlled information, an `<UnclassifiedControlled>` element must be present with a `<Type>` element containing this text in its element content with a single blank character separating all words:

```
Export Controlled Information
```

The following text must appear in `<Admonishment>` element content:

Contains technical data whose export is restricted by statute. Violations may result in administrative, civil, or criminal penalties. Limit dissemination to U.S. Department of Energy and major U.S. DOE contractors. The cognizant program manager must approve other dissemination. This notice shall not be separated from the attached document.

The name of the reviewer is placed in `<Reviewer>` element content and the date the review was performed is placed in `<DateReviewed>` element content.

C.2 NAVAL NUCLEAR PROPULSION INFORMATION

Naval Nuclear Propulsion Information is information about the design, arrangement, development, manufacture, testing, operation, administration, training, maintenance, or repair of the propulsion plants of naval nuclear-powered ships and prototypes, including the associated nuclear support facilities.

If the information package contains Naval Nuclear Propulsion information, an `<UnclassifiedControlled>` element must be present with a `<Type>` element containing this text in its element content with a single blank character separating all words:

```
Naval Nuclear Propulsion Information
```

The following text must appear in `<Admonishment>` element content:

NOFORN. This document is subject to special export controls and each transmittal to foreign governments or foreign nationals must be made only with the prior approval of the NavSea.

C.3 SENSITIVE NUCLEAR TECHNOLOGY

Sensitive Nuclear Technology includes any information, and only that information, that is not Restricted Data, not available to the public, and "important" to the design, construction, operation, or maintenance of a facility for uranium enrichment, nuclear fuel reprocessing, or heavy water production.

If the information package contains Sensitive Nuclear Technology information, an `<UnclassifiedControlled>` element must be present with a `<Type>` element containing this text in its element content with a single blank character separating all words:

```
Sensitive Nuclear Technology
```

The following text must appear in `<Admonishment>` element content:

SPECIAL HANDLING REQUIRED NOT RELEASABLE TO FOREIGN NATIONALS

C.4 APPLIED TECHNOLOGY

Applied Technology (AT) is an unclassified category of information established by the Office of Nuclear Energy, Science, and Technology (DOE/NE) to preserve the foreign trade value of certain DOE/NE-funded progress and topical reports containing engineering, development, design, construction, and operation information pertaining to particular programs.

If the information package contains Applied Technology information, an `<UnclassifiedControlled>` element must be present with a `<Type>` element containing this text in its element content with a single blank character separating the words:

```
Applied Technology
```

The following text must appear in the <Admonishment> element content:

APPLIED TECHNOLOGY Any further distribution by any holder of any document or data therein to third parties representing foreign interests, foreign governments, foreign companies, and foreign subsidiaries or foreign divisions of U.S. companies shall be approved by the (insert appropriate NE program office officials), U.S. Department of Energy. Further, foreign party release may require DOE approval pursuant to Federal Regulation 10 CFR Part 810, and/or may be subject to Section 127 of the Atomic Energy Act.

The specific Office of Nuclear Energy, Science, and Technology positions must be specified.

C.5 COOPERATIVE RESEARCH AND DEVELOPMENT AGREEMENT

Cooperative Research and Development Agreement (CRADA) information is data produced in the performance of the agreement that would have been proprietary information had it been obtained from a non-federal entity.

If the information package contains CRADA information, an <UnclassifiedControlled> element must be present with a <Type> element containing this text in its element content with a single blank character separating all words:

```
PROTECTED CRADA INFORMATION
```

The following text must appear in <Admonishment> element content with the CRADA number in (number):

PROTECTED CRADA INFORMATION This product contains Protected CRADA Information, which was produced on (date) under CRADA No. (number) is not to be further disclosed for a period of five years from the date it was produced except as expressly provided for in the CRADA.

C.6 CONFIDENTIAL/FOREIGN GOVERNMENT INFORMATION–MODIFIED HANDLING

Confidential/Foreign Government Information—Modified Handling (C/FGI-MOD) is a controlled information type used to protect foreign government information that must be protected at a level below the level of protection required for information classified Confidential. Assignment to this category can only be performed by a derivative classifier.

If the information package contains C/FGI-MOD information, an <UnclassifiedControlled> element must be present with a <Type> element containing this text in its element content with a single blank character separating all words:

```
Confidential/Foreign Government Information—Modified Handling.
```

The / and – characters must not have space characters between them and the words they separate.

The following text must appear in <Admonishment> element content:

This document contains (insert name of country) (insert classification level) information to be treated as Confidential—Modified Handling Authorized.

The name of the country and classification level must be specified.

The information on how the information type assignment was determined is placed in the <ClassifiedBy> and <DerivedFrom> elements. These element names are the same used to identify the derivative classifier in the <Classification> element.

The foreign government classification markings are placed in <AdditionalInformation> element content.

C.7 CONTRACTOR-OWNED INFORMATION

Contractor-owned information is that which is generated by the contracting company for its use only and needs to be protected from unauthorized disclosure.

If the information package contains Y-12 contractor-owned information, an <UnclassifiedControlled> element must be present with a <Type> element containing this text in its element content with a single blank character separating all words:

```
Contractor-Owned Information
```

The – character must not have space characters between them and the words it separates.

The <Admonishment> element must be present and contain one of these statements in its element content:

COMPANY USE ONLY This document contains privileged information and must be protected from disclosure outside the company unless release is authorized by the Legal Division.

BUSINESS PERSONAL This document contains administrative or employee-sensitive information that must be protected from disclosure to those without a need to know. Disclosure outside the company is prohibited unless release is authorized by the Legal Division.

COMPANY SENSITIVE This document contains administrative or employee-sensitive information that must be protected from disclosure to those without a need to know. Disclosure outside the company is prohibited unless release is authorized by the Legal Division.

C.8 PRIVACY ACT INFORMATION

Privacy Act information consists of certain types of information about individuals such as information that can be used to identify a person, payroll information, medical information, etc.

If the information package contains Privacy Act information, an `<UnclassifiedControlled>` element must be present with a `<Type>` element containing this text in its element content with a single blank character separating all words:

```
Privacy Act Information
```

The `<Admonishment>` element must be present and contain this statement in its element content:

PRIVACY ACT INFORMATION RESTRICTIONS ON DISCLOSURE This record contains personal/confidential information and is subject to protection by the Privacy Act of 1974; 5 U.S.C. Sect. 552(a). Federal or contractor employees and their subcontractors who willfully make an unauthorized disclosure of information from this record shall be guilty of a misdemeanor and fined up to $5,000.

The name and organization of the person that made the determination is placed in `<Reviewer>` element content and the date the determination was made is placed in `<DateReviewed>` element content.

APPENDIX D

OAIS REFERENCE MODEL CONFORMANCE

The information package described in this specification conforms to the description of an information package in *Reference Model for an Open Archival Information System (OAIS)* [OAIS1, OAIS2]. This Reference Model describes a possible architecture of a complete archival information system designed to receive information to be archived, manage that information in an archive, and provide that information to users. It defines the concepts and terms that can be used to specify the components and processes of an information archive. The OAIS Reference Model is being used to develop archiving systems by the National Archives and Records Administration, Library of Congress, National Aeronautics and Space Administration, and by many other public and private organizations.

This Reference Model was developed by the Consultative Committee for Space Data Systems (CCSDS), an organization officially established by the management of its member space agencies. This committee meets periodically to address data systems problems that are common to all participants and to formulate sound technical solutions to these problems. It issues Recommendations that document the technical solutions it identifies. This Reference Model document is a CCSDS Recommendation.

From the Reference Model introduction [OAIS1]:

> *The purpose of this document is to define the International Organization for Standardization (ISO) Reference Model for an Open Archival Information System (OAIS). An OAIS is an archive, consisting of an organization of people and systems, that has accepted the responsibility to preserve information and make it available for a Designated Community. It meets a set of such responsibilities as defined in this document, and this allows an OAIS archive to be distinguished from other uses of the term 'archive'. The term 'Open' in OAIS is used to imply that this Recommendation, as well as future related Recommendations and standards, are developed in open forums, and it does not imply that access to the archive is unrestricted*

This Reference Model has been accepted by the International Organization for Standardization (ISO) and is ISO standard ISO 14721:2003.

The information package described in this specification is an implementation of the OAIS Reference Model archival information package tailored to meet NNSA information storage requirements and the computing environment used at Y-12 and other NSE sites. Information packages created using this specification can be stored and managed using OAIS-compliant archive systems.

D.1 OAIS ENVIRONMENT

The OAIS Reference Model describes an open archival information system that operates in an environment of *Producers* that provide information to be stored in an *Archive* and *Consumers* that use the information stored in the *Archive*. The Reference Model describes these roles as follows:

***Producer**: The role played by those persons, or client systems, who provide the information to be preserved. This can include other OAISs or internal OAIS persons or systems.*

***Archive**: An organization that intends to preserve information for access and use by a Designated Community.*

***Consumer**: The role played by those persons, or client systems, who interact with OAIS services to find preserved information of interest and to access that information in detail. This can include other OAISs, as well as internal OAIS persons or systems.*

The *Consumer* role includes a *Designated Community* with a *Knowledge Base*:

***Designated Community**: An identified group of potential Consumers who should be able to understand a particular set of information. The Designated Community may be composed of multiple user communities.*

***Knowledge Base**: A set of information, incorporated by a person or system, that allows that person or system to understand received information.*

This information package specification assumes the roles are filled as follows:

Producers are the Y-12 Quality, Surveillance, and other activities that generate the information stored in information packages.

Archive is the Windchill PDMLink, Product Characterization System, and SAP systems used to manage information generated by the Producers.

Consumers and *Designated Community* are the Y-12 and NSE Design Agency persons who must have access to the information in the *Archive* to perform their work. These *Consumers* are assumed to have an extensive *Knowledge Base* that, for example, includes the procedures, specifications, and other information required to understand a set of quality data.

D.2 INFORMATION PACKAGE

The OAIS Reference Model uses this definition of an *Information Package*:

*An **Information Package** is a conceptual container of two types of information called Content Information and Preservation Description Information (PDI). The Content Information and PDI are viewed as being encapsulated and identifiable by the Packaging Information. The resulting package is viewed as being discoverable by virtue of the Descriptive Information.*

The OAIS Reference Model defines *Content Information* as:

*The **Content Information** is that information which is the original target of preservation. It consists of the Content Data Object (Physical Object or Digital Object, i.e., bits) and its associated Representation Information needed to make the Content Data Object understandable to the Designated Community.*

***Content Data Object**: The Data Object, that together with associated Representation Information, is the original target of preservation.*

***Data Object**: Either a Physical Object or a Digital Object.*

***Physical Object**: An object (such as a moon rock, bio-specimen, microscope slide) with physically observable properties that represent information that is considered suitable for being adequately documented for preservation, distribution, and independent usage.*

***Digital Object**: An object composed of a set of bit sequences.*

Package information is *Content Information* in this specification. It is information such as inspection information and the context required to understand and use that information. Package information consists only of digital objects; physical objects (such as moon rocks) cannot be represented by an information package.

Representational Information consists of *Structure Information* or *Syntactic Information* and *Semantic Information*. XML element tags are expected to provide much of the *Structure Information* for information stored in information packages. For example, a weight may be stored as follows:

```
<Weight units=”kg”>25</Weight>
```

The element tag name identifies the information and the units attributes specifies the units of measure. The *Designated Community* that will be the users of this information share a common *Knowledge Base*, so *Semantic Information* is not addressed in this specification. This specification includes recommendations that formatted or other structured information either include a description of the information or a reference to external information that describes the information.

D.3 PRESERVATION DESCRIPTION INFORMATION

The OAIS Reference Model defines *Preservation Description Information* as:

Preservation Description Information (PDI)*: The information which is necessary for adequate preservation of the Content Information and which can be categorized as Provenance, Reference, Fixity, and Context information.*

The mapping between the terms in this definition and this information packages specification are below. Following each OAIS Reference Model concept in indented text is how the information package described in this specification implements the concept.

Provenance Information*: The information that documents the history of the Content Information. This information tells the origin or source of the Content Information, any changes that may have taken place since it was originated, and who has had custody of it since it was originated. Examples of Provenance Information are the principal investigator who recorded the data, and the information concerning its storage, handling, and migration.*

This OAIS concept is represented in this specification by the package metadata `<History>` and `<Notes>` elements and by information-specific elements in the package information.

Reference Information*: The information that identifies, and if necessary describes, one or more mechanisms used to provide assigned identifiers for the Content Information. It also provides identifiers that allow outside systems to refer, unambiguously, to a particular Content Information. An example of Reference Information is an ISBN*

Package metadata includes the `<PackageIdentifier>` element that includes a unique identifier for the information package, an `<AlternateIdentifier>` element that can contain other identifiers associated with the package information, and other information that can be used to identify the package. Package information is expected to include additional identifiers that identify the information.

Fixity Information*: The information which documents the authentication mechanisms and provides authentication keys to ensure that the Content Information object has not been altered in an undocumented manner. An example is a Cyclical Redundancy Check (CRC) code for a file.*

XML Signatures provide this capability for information packages described by this specification.

Context Information*: The information that documents the relationships of the Content Information to its environment. This includes why the Content Information was created and how it relates to other Content Information objects.*

The relationships between package information and the environment are expected to be included in the package information. Relationships between information packages can be specified using the package

metadata <References> element and additional information such as why the information package was created can be in the <Notes> and <History> metadata elements.

D.4 PACKAGING AND DESCRIPTIVE INFORMATION

These packaging and descriptive information concepts are defined in the OAIS Reference Model:

Packaging Information *is that information which, either actually or logically, binds, identifies and relates the Content Information and PDI.*

Descriptive Information *is that information which is used to discover which package has the Content Information of interest. Depending on the setting, this may be no more than a descriptive title of the Information Package that appears in some message, or it may be a full set of attributes that are searchable in a catalog service.*

The information package XML structure binds the *Content Information* contained in the <PackageInfo> element and the *Preservation Description Information* contained in package metadata elements. *Descriptive Information* is provided by the information package <PackageDescription>, <SearchTerms>, and <AccessControl> elements.

The Reference Model defines the *Package Description* as

Package Description*: The information intended for use by Access Aids.*

Access Aids are OAIS components used to manage information packages. Package description information is stored by the OAIS separately from the information package and used by the OAIS to locate and return information for a *Consumer*. These package metadata elements contain information that can be used to create an OAIS *Package Description*:

- <PackageIdentification> – Package identifiers, description, status, and dates
- <InformationMarking> – Information classification and sensitivity
- <AccessControl> – Access control attributes
- <SearchTerms> – Search terms

D.5 INFORMATION PACKAGE TYPES

The OAIS Reference Model describes three types of OAIS *Information Packages*:

Submission Information Package (SIP)*: An Information Package that is delivered by the Producer to the OAIS for use in the construction of one or more AIPs*

Archival Information Package (AIP)*: An Information Package, consisting of the Content Information and the associated Preservation Description Information (PDI), which is preserved within an OAIS*

Dissemination Information Package (DIP)*: The Information Package, derived from one or more AIPs, received by the Consumer in response to a request to the OAIS*

The information package defined in this specification is an *Archival Information Package* that is used as the *Submission Information Package* and *Dissemination Information Package*. It is expected to be created in its archival form by the *Producer* before delivery to the system used to store it and to be stored in this system as an *Archival Information Package*. While being stored, information package metadata may be altered as required to manage the information package or the OAIS. The archival package is provided directly to the *Consumer*.

D.6 ADDED CONCEPTS

This information package extends the OAIS Information Package by adding information marking and access control. All information generated by the Department of Energy must be marked as required to protect the information. This information package specification provides for all current DOE information markings. Access to information must further be limited to persons with a need-to-know the information. The OAIS Reference Model allows for access control but does not describe how access is controlled. This information package adds an `<AccessControl>` element that contains the information required to control access to the information in the information package.

D.7 IMPLEMENTATION ISSUES

All but a few OAIS *Information Package* concepts are implemented by the information package defined in this specification. Many concepts are implemented both in information package metadata and in package information. For example, *Provenance* information can be in package metadata and package information.

The OAIS Reference Model allows complex *Information Objects* that can themselves contain complex *Information Objects* in all of its concepts. Package metadata are implemented as simple text strings whose *Representational Information* consists of containing element tag names and attributes. If complex *Information Objects* are required for information normally in package metadata, this information must be placed instead in the package information.

Package information is an *Information Object* that can contain *Information Objects*. However, most *Information Objects* in package information are also expected to be implemented as simple text strings whose *Representational Information* consists of containing element tag names and attributes.

In the OAIS Reference Model, Packaging Information is independent of the *Archival Information Package*. Information packages defined by this specification are designed to be managed as a single entity

and to be independent of the system that stores them, so the information package includes the *Packaging Information*. This *Packaging Information* is the information package XML structure.

Every information package is an *Archival Information Package* and is organized as an *Archival Information Unit*, defined as:

> ***Archival Information Unit (AIU)****: An Archival Information Package whose Content Information is not further broken down into other Content Information components, each of which has its own complete Preservation Description Information. It can be viewed as an 'atomic' AIP.*

The Reference Model allows for *Archival Information Collections*, defined as:

> ***Archival Information Collection (AIC)****: An Archival Information Package whose Content Information is an aggregation of other Archival Information Packages.*

The information package defined by this specification is not designed to be a container for other information packages and it is not expected to be used in this way. The `<References>` element can be used to create a collection of information packages by placing the package identifiers of the collection member information packages in `<Reference>` elements.

D.8 NOT ADDRESSED

The OAIS Reference Model describes a complete architecture for an Open Archival Information System that consists of an information system that receives *Submission Information Packages* from *Producers*, stores them as *Archival Information Packages*, and provides *Dissemination Information Packages* when requested to authorized *Consumers*. This information package specification implements only the OAIS *Information Package* concepts described in the Reference Model. These parts of the OAIS Reference Model are not addressed by this specification:

- Management and administration of the Open Archival Information System
- Creation and administration of information package collections
- Administration of information packages and related information
- Availability of external information referenced in information packages
- Relationship between the *Producer* and OAIS management and administration
- Relationship between the *Consumer* and OAIS management and administration
- Access aids used to locate and retrieve information packages

These aspects of information package management and use are or will be considered by other information management projects.

INTERNAL DISTRIBUTION

J. B. Atwater

D. R. Baumgardner

A. L. Burton

R. L. Crisp

S. E. Hughes

T. M. Insalaco

R. A. Lewis

R. L. Luttrell

M. D. Love

M. A. McNeil

P. M. Parris

C. H. Richter

B. K. Robinette

R. C. Secrist

K. J. Smith

T. O. G. Tallant

R. M. Wilson

EXTERNAL DISTRIBUTION

D Bellis, OSTI

D. R. Hamrin, ORNL

R. Harris, SNL

K. E. Langley, SAIC

B. E. Lownsbury, LLNL

M. A. Lane, LLNL

A. E. Russell, LLNL

R. L. Shoup, LANL