GUIDE TO OSCAL-BASED FEDRAMP CONTENT

Version 1.0

June 1, 2020

FedRAMP

DOCUMENT REVISION HISTORY

Date	Description	Version	Author
6/1/2020	Initial Publication	1.0	FedRAMP PMO
<Date>	<Revision Description>	<Version>	<Author>
<Date>	<Revision Description>	<Version>	<Author>

How to Contact Us

For questions about FedRAMP, or for technical questions about this document including how to use it, contact *info@FedRAMP.gov.*

For more information about FedRAMP, see https://FedRAMP.gov.

TABLE OF CONTENTS

1. OVERVIEW

1.1. Who Should Use This Document?

This document is intended for technical staff and tool developers implementing solutions for importing, exporting, and manipulating Open Security Controls Assessment Language (OSCAL)-based FedRAMP content, such as system security plans (SSP), security assessment plans (SAP), security assessment reports (SAR), and plans of action and milestones (POA&M).

It provides guidance and examples for an organization producing and using OSCAL-based, FedRAMP-compliant files. Our goal is to enable your organization to develop tools that will seamlessly ensure these standards are met so your security practitioners can focus on content and accuracy rather than formatting and presentation.

1.2. Related Documents

This document does not stand along. It provides foundational information and core concepts, which apply to the following four guides:

- *Guide to OSCAL-based FedRAMP System Security Plans (SSP)*
- *Guide to OSCAL-based FedRAMP Security Assessment Plans (SAP)*
- *Guide to OSCAL-based FedRAMP Security Assessment Reports (SAR)*
- *Guide to OSCAL-based FedRAMP Plan of Action and Milestones (POA&M)*

Each of those documents will refer back to this one for common concepts.

1.3. Basic Terminology

XML and JSON use different terminology. Instead of repeatedly clarifying format-specific terminology, we use the following format-agnostic terminology through these documents.

TERM	XML EQUIVALENT	JSON EQUIVALENT
`Field`	A single element or node that can hold a value or an attribute	A single object that can hold a value or property
`Flag`	Attribute	property
`Assembly`	A collection of elements or nodes. Typically, a parent node with one or more child nodes.	A collection of objects. Typically, a parent object with one or more child objects.

These terms are used by the National Institute of Standards and Technology (NIST) in the creation of OSCAL syntax.

Throughout these documents, the following words are used to differentiate between requirements, recommendations, and options.

TERM	MEANING
must	Indicates a requirement.
should	Indicates a recommendation, which is not necessarily required.
may	Indicates an available optional, which may be used at the organization's discretion.

1.4. XML and JSON Formats

The examples provided here are in XML; however, FedRAMP accepts XML or JSON formatted OSCAL content. NIST offers the ability to convert OSCAL-files between XML and JSON in either direction without data loss.

You may submit your SSP, SAP, SAR, and POA&M to FedRAMP using either XML or JSON. If necessary, FedRAMP's tools will convert the files for processing.

For more information on converting OSCAL files between XML and JSON, see Section 1.6.2 NIST OSCAL Format Conversion Mechanisms.

1.5. OSCAL-based FedRAMP Templates

FedRAMP offers OSCAL-based templates in both XML and JSON formats for the SSP, SAP, SAR, and POA&M. These templates contain many of the FedRAMP required content and placeholders to help get you started. This document is intended to work in concert with those templates. The OSCAL-based FedRAMP templates are available here:

- https://github.com/GSA/fedramp-automation/raw/master/templates

1.6. XML and JSON Technology Standards

For OSCAL compliance, mechanisms that interpret or generate OSCAL content must honor the core syntax described at https://pages.nist.gov/OSCAL/documentation/schema/.

While not mandatory, organizations adopting OSCAL are strongly encouraged to use the NIST-published validation and translation mechanisms. The validation mechanism ensures XML and JSON files are using OSCAL-compliant syntax, while the translation mechanism converts OSCAL content from either format to the other. NIST has an automated governance process, which ensures these mechanisms remain aligned with the latest OSCAL syntax.

***TIP**: There are comments in the XML versions of the FedRAMP Templates. Unfortunately, JSON does not formally support comments. JSON users may wish to review the comments in the equivalent sections of the XML files.*

1.6.1. NIST OSCAL Syntax Validation Mechanisms

The latest version of NIST OSCAL schema validation files are always available here:
XML: https://github.com/usnistgov/OSCAL/tree/master/**xml**/schema
JSON: https://github.com/usnistgov/OSCAL/tree/master/**json**/schema

Validating XML-based OSCAL files using the NIST-published schema validation requires:
XML Schema Definition Language (XSD) 1.1
[https://www.w3.org/TR/xmlschema11-1/]

Validating JSON-based OSCAL files using the NIST-published schema validation requires:

JSON Schema, draft-07
[https://json-schema.org/]

There are several open-source and commercial tools that will process XSD 1.1 or JSON Schema, draft-07, either as stand-alone capabilities or as programming libraries.
FedRAMP and NIST are unable to endorse specific products.

1.6.2. NIST OSCAL Format Conversion Mechanisms

The latest version of NIST OSCAL format conversion files are always available here:

XML to JSON: https://github.com/usnistgov/OSCAL/tree/master/**json**/convert
JSON to XML: https://github.com/usnistgov/OSCAL/tree/master/**xml**/convert

Converting between XML and JSON in either direction using the NIST built the XML conversion files requires:

Extensible Stylesheet Language Transformation (XSLT) 3.0
[https://www.w3.org/TR/2017/REC-xslt-30-20170608/]

and:

XPath 3.1
[https://www.w3.org/TR/xpath-31/]

There are several open-source and commercial tools that will process XSLT3.0 and XPath 3.1, either as stand-alone capabilities or as programming libraries.
FedRAMP and NIST are unable to endorse specific products.

1.7. XPath Queries and References

Except where noted, all XPath queries in this document are based on XPath 2.0.

Most modern programming languages make XPath 1.0 available by default. XPath 2.0 can typically be added with third-party libraries, or calls to external command-line utilities.

The XPath queries *in this document* have been designed to easily convert from XPath 2.0 to 1.0 using a function similar to the one below, which enables developers to use native XML capabilities.

Sample PHP Code to Prepend Namespace (Converts *most* XPath 2.0 to 1.0)

```
// $query is the XPath 2.0 query
// $ns is the namespace value which much be prepended to element
//      names in the query
// RETURNS: An XPath 1.0 query
function AddNamespace2xpath($query, $ns) {
      $result = "";
      $q_len = strlen($query);
      $prev_char = "";

      for($i=0; $i < $q_len ; $i++) {
            $cur_char = substr($query, $i, 1);
            if ($prev_char === '/') {
                  if (ctype_alpha($cur_char)) {
                        $result .= $ns . ":";
                  };
            }
            $result .= $cur_char;
            $prev_char = $cur_char;
      }

      return $result;
}
```

Even if you do not use XPath to query the data file, the XPath queries provide a concise and non-ambiguous way to communicate where the data is within the file.

Most XPath queries in this document are absolute paths from the root of the document. In other words, it is clear from the XPath query which of the major file sections described in Section 2.3 is being referenced, and where the field is located within the section.

***JSON Users**: There are several JSON query technologies available, such as JSONPath [https://restfulapi.net/json-jsonpath/]; however, no one technology has emerged as a clear standard as of this publication.*

***Database Users**: Some tool developers prefer to manage OSCAL data by first importing it into a database. This is an acceptable approach, provided any resulting OSCAL file generated form the database follows the OSCAL syntax. If the file is intended for delivery to FedRAMP, it must also follow the guidance in this document.*

There are also database-like XML servers, such as the open-source tool BaseX [http://www.basex.org/], which allow OSCAL files to remain in their native format, yet be queried more like a traditional database. The queries are also optimize for performance.

2. WORKING WITH OSCAL FILES

This section covers several important concepts and details that apply to OSCAL-based FedRAMP files.

2.1. Hierarchy and Sequence

For both XML and JSON, the hierarchy of syntax is important and must be followed. For example, the `metadata` assembly must be at the top level of the OSCAL file, just below its root. If it appears within any other assembly, it is invalid.

The same field name is interpreted differently in different assemblies. For example, the `title` field under metadata is the document title, while the `title` field under role gives a human-friendly name to that role.

For XML, the sequence of syntax within an assembly is also important; however, JSON is flexible.

For example, within the `metadata` assembly, XML must find `title`, `published`, `last-modified`, `version`, and `oscal-version` in exactly that order. The published field is optional and may be omitted, but if present, it must be in that position relative to the other fields. When using JSON, these fields are allowed in any order within the `metadata` assembly.

Fortunately, NIST's documentation presents the fields and assemblies in the correct sequence.

Always use the hierarchy found here:

- SSP: https://pages.nist.gov/OSCAL/documentation/schema/ssp/xml-model-map/
- SAP: https://pages.nist.gov/OSCAL/documentation/schema/assessment-plan/xml-model-map/
- SAR: https://pages.nist.gov/OSCAL/documentation/schema/assessment-results/xml-model-map/
- POA&M: https://pages.nist.gov/OSCAL/documentation/schema/poam/xml-model-map/

2.1.1. Typical OSCAL Assembly Structure

Most assemblies in OSCAL follow a general pattern as follows:

- `title` (field - typically one, sometimes zero or one)
- `description` (field - typically one, sometimes zero or one)
- `prop` (fields - zero or more - also used to extend OSCAL)
- `annotation` (assemblies - zero or more - also used to extend OSCAL)
- ***assembly-specific fields***
- `remarks` (field - zero or one)

While this is not universal in OSCAL, when any of these fields are present, they follow this pattern.

2.2. Multiple Layers of Validation

There are several layers at which an OSCAL file can be considered valid.
FedRAMP requires all layers be satisfied.

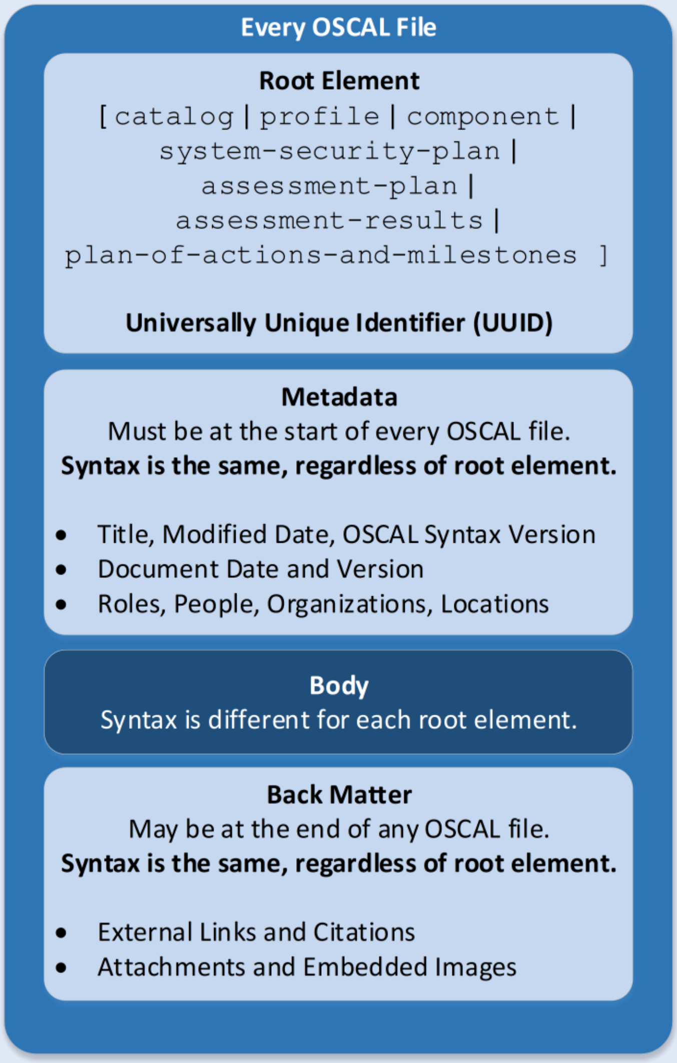
LAYER	DESCRIPTION
Well-Formed	The XML or JSON file follows the rules defined for that format. Any tool that processes the format will recognize it as "well-formed," which means the tool can proceed with processing the XML or JSON. XML: https://www.w3.org/TR/REC-xml/ JSON: https://json.org/
OSCAL Syntax	The XML or JSON file only uses tagging that is defined by NIST for OSCAL. NIST publishes schema validation tools to verify syntax compliance based on the following standards: XML Syntax Validation: XML Schema Definition Language (XSD) 1.1 JSON Syntax Validation: JSON Schema, draft 07
OSCAL Content	Currently NIST only enforces certain declarations using the validation files above. This means defining a limited set of acceptable values for certain fields. For example, NIST declared the following as the only acceptable values for the impact levels for information types: `fips-199-low`, `fips-199-moderate`, and `fips-199-high` within the SSP syntax. Any other value will cause an error when validating the file using the NIST syntax validation capabilities above. In the future, NIST intends to publish content enforcement mechanisms, such as Schematron. This will enforce more complex rules such as, "If field A is marked 'true', field B must have a value." FedRAMP is also exploring the use of Schematron for enhanced validation, and may publish these in the future.
FedRAMP Syntax Extensions	NIST built a language they believe represents the commonality of most cybersecurity frameworks and provided the ability to extend the language for framework-specific needs. FedRAMP makes use of these extensions. NIST provides `prop` and `annotation` fields throughout most of its assemblies, always with a `name`, `class`, and `ns` (namespace) flag: `<prop name="" class="" ns="">Data</prop>` In the core OSCAL syntax, the `ns` flag is never used. Where FedRAMP extends OSCAL, the value for `ns` is always: `https://fedramp.gov/ns/oscal` (case sensitive). When `ns='https://fedramp.gov/ns/oscal'` the `name` flag is as defined by FedRAMP. If the `class` flag is present, that is also defined by FedRAMP.
FedRAMP Content	Today, FedRAMP content is enforced programmatically. FedRAMP will evaluate the use of Schematron (above) for future FedRAMP validation efforts.

2.3. OSCAL's Minimum File Requirements

Every OSCAL-based FedRAMP file must have a minimum set of required fields/assemblies, and must follow the core OSCAL syntax found here:

https://pages.nist.gov/OSCAL/documentation/schema/

In addition to the core OSCAL syntax, the following FedRAMP-specific implementation applies:



Every OSCAL File

Root Element
[`catalog` | `profile` | `component` | `system-security-plan` | `assessment-plan` | `assessment-results` | `plan-of-actions-and-milestones`]

Universally Unique Identifier (UUID)

Metadata
Must be at the start of every OSCAL file.
Syntax is the same, regardless of root element.

- Title, Modified Date, OSCAL Syntax Version
- Document Date and Version
- Roles, People, Organizations, Locations

Body
Syntax is different for each root element.

Back Matter
May be at the end of any OSCAL file.
Syntax is the same, regardless of root element.

- External Links and Citations
- Attachments and Embedded Images

Anatomy of an OSCAL-based file

- The root element of the file indicates the type of content within the body of the file. The recognized OSCAL root element types are as follows:
  - **catalog**: Contains a catalog of control definitions, as well as related assessment goals and activities, such as NIST SP 800-53 and 800-53A.
  - **profile**: Contains a control baseline, such as FedRAMP Moderate
  - **component**: Contains information about a product, service, or other security capability, such as the controls it can satisfy and what settings to use to meet the requirements of a particular baseline.
  - **system-security-plan**: Describes a system and its security capabilities, including its authorization boundary and a description of how each control is satisfied.
  - **assessment-plan**: Describes the plan for assessing a specific system.
  - **assessment-results**: Describes the actual activities performed in the assessment of a specific system, as well as the results of those activities.
  - **plan-of-actions-and-milestones**: Describes and tracks known vulnerabilities of a system, as well as the plan for remediation.
- The Universally Unique ID (UUID) at the root must be changed every time the content of the file is modified.
- Every OSCAL file must have a metadata section and include a title, last-modified date, and OSCAL syntax version.

The table below shows an empty OSCAL file, based purely on the NIST syntax; however, FedRAMP requires much more in a minimum file. The latest OSCAL-based FedRAMP template files can be found here in JSON and XML formats:

https://github.com/GSA/fedramp-automation/tree/master/templates

An Empty OSCAL File Representation

```
<?xml version="1.0" encoding="UTF-8"?>
<OSCAL-root-element xmlns="http://csrc.nist.gov/ns/oscal/1.0"
   id="[generated-uuid]">
   <metadata>
      <title>Document Title</title>
      <last-modified>2019-11-27T00:00:00.00-05:00Z</last-modified>
      <version>0.0</version>
      <oscal-version>1.0-Milestone3</oscal-version>
   </metadata>

   <!-- body cut -->

   <back-matter />
</OSCAL-root-element>
```

2.3.1. UTF-8 Character Encoding

OSCAL uses UTF-8 character encoding. JSON files are always UTF-8 character encoded.

In XML, the first line in the example above ensures UTF-8 encoding is used. Other encoding options will create unpredictable results.

2.3.2. OSCAL Syntax Version

Tools designed to read an OSCAL file must verify the `oscal-version` field to determine which published syntax is used.

Tools designed to create or manipulate an OSCAL file must specify the syntax version of OSCAL used in the file in the `oscal-version` field.

NIST ensures backward compatibility of syntax where practical; however, this is not always possible. Some syntax changes between milestone releases leading up to OSCAL version 1.0 are unavoidable. NIST intends to keep all formally published schema validation files available, which keeps validation and conversion tools available for older versions of OSCAL. See Section *2.3.6 OSCAL Syntax Versions* for more information.

2.3.3. OSCAL File Changes

Any time a tool changes the contents of an OSCAL file, it must also:

- update the file's `id` flag (`/[OSCAL-root-element]/@id`) with a new UUID; and
- update the `last-modified` field (`/*/metadata/last-modified`) with the current date and time. (Using the OSCAL date/time format, as described in *Section 2.5.1 Date and Time in OSCAL Files*)

Tools that work with updates to an OSCAL file should rely on the UUID value provided by the `id` flag, and `last modified` field as easy methods of knowing the file has changed.

2.3.4. Cryptographic Integrity (Future)

NIST intends to add a cryptographic hash feature to OSCAL during calendar year 2020. Once added by NIST, FedRAMP will update this section.

While tool developers are encouraged to perform their own integrity checking, it is important to note cryptographic hash algorithms will produce a different result for inconsequential file differences, such as different indentation or a change in the sequence of flags.

2.3.5. Useful XPath Queries for Document Changes and OSCAL Syntax

Below are a few important queries, which enable a tool to obtain critical information about any OSCAL file.

XPath Queries

```
OSCAL syntax version used in this file:
  /*/metadata/oscal-version

Last Modified Date/Time:
  /*/metadata/last-modified

Unique Document ID:
  /*/@id

Document Title:
  /*/metadata/title
```

2.3.6. OSCAL Syntax Versions

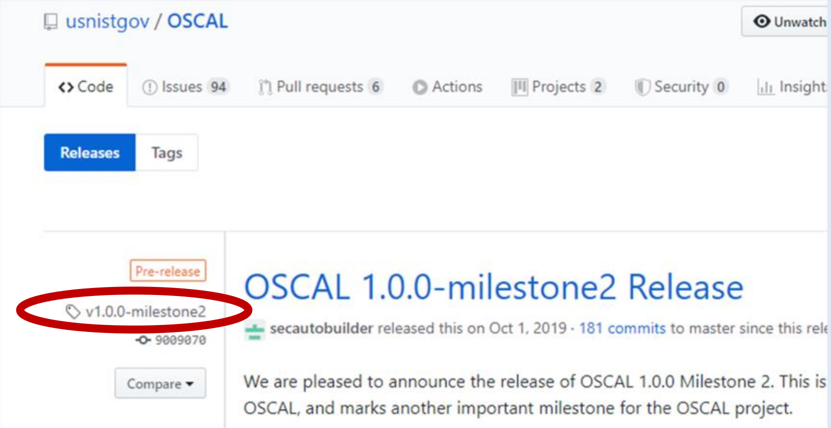
NIST's approach to OSCAL development ensures the syntax validation and format conversion tools remain available for each milestone and full release of OSCAL. As of this publication, the following syntax versions and associated resources are or will be made available:

SYNTAX VERSION	RESOURCES
Version 1.0.0 Full Release Identifier: `1.0.0`	*Expected August/September 2020*
Version 1.0.0 Pre-Release Milestone #3 Identifier: `1.0.0-milestone3`	*Expected June 2020* XML Schema Validation JSON Schema Validation XML to JSON Conversion JSON to XML Conversion *Links above will not work until NIST publishes MS3.*
Version 1.0.0 Pre-Release Milestone #2 Identifier: `1.0.0-milestone2`	**Published** October 1, 2019 XML Schema Validation JSON Schema Validation XML to JSON Conversion JSON to XML Conversion
Version 1.0.0 Pre-Release Milestone #1 Identifier: `1.0.0-milestone1`	**Published** June 15, 2019 XML Schema Validation JSON Schema Validation XML to JSON Conversion JSON to XML Conversion

NIST always makes the latest version of syntax validation and format conversion files available in the master OSCAL repository, including any changes since the last formal release.

To ensure stable resources, use a formal OSCAL release. NIST publishes formal OSCAL releases here: https://github.com/usnistgov/OSCAL/releases

To access OSCAL resources based on a particular release, click the tag to the left of the release title.



OSCAL 1.0.0-milestone2 Release

secautobuilder released this on Oct 1, 2019 · 181 commits to master since this rele

We are pleased to announce the release of OSCAL 1.0.0 Milestone 2. This is OSCAL, and marks another important milestone for the OSCAL project.

2.4. Assigning Identifiers

There are three ways identifiers are typically used in OSCAL:

- **ID flag**: uniquely identifies a field or assembly.
- **ID Reference**: a flag or field that points to another field or assembly using its ID unique flag value.
- **Uniform Resource Identifier (URI) Fragment**: A value in a flag or field with a URI data type. A URI fragment starts with a hashtag (#) followed by a unique ID value.

ID flags appear as an "**id**" flag to a data field or assembly. Examples include:

- `<interconnection` **`id`**`="ic-001">`: Uniquely identifies the interconnection
- `<party` **`id`**`="csp">`: Uniquely identifies the party

IDs are intended to be managed by tools "behind the scenes," and should not typically be exposed to users.

Throughout the OSCAL syntax, if an ID flag is available, it is typically required.

An ID reference typically appears with a name and hyphen in front of the "id" (name-id). It is typically a flag, but sometimes a field. The name of an ID reference flag/field typically reflects the name of the field to which it points.

Examples include:

- `<responsible-party` **`role-id`**`="prepared-by">`: points to a `role` identified by "prepared-by".
- `<implemented-requirement id="imp-req-01"` **`control-id`**`="ac-2">`: points to the `control` identified by "ac-2".

NIST provides some standard identifiers. Where appropriate, FedRAMP has adopted those and defined additional identifiers as needed. To ensure consistent processing, FedRAMP encourage content creators to use the NIST and FedRAMP-defined identifiers to the greatest degree practical. Deviation is likely to result in processing errors.

2.4.1. Uniqueness of Identifiers

OSCAL syntax validation tools currently require an identifier to be unique among like-named fields. For example, the same ID value must not be present for two different `role` fields; however, the ID value for a `party` field may be the same as the ID value for a `role` field without conflict.

As a best practice, the FedRAMP Program Management Office (PMO) recommends ensuring IDs are unique within the file as a whole and not just within the scope of like-named elements.

This is especially important if using URI fragments in URI fields such as `href="#id-value"`. NIST does not yet have clear rules for handling URI fragments when the referenced id-value appears more than once in the file. Keeping IDs unique within the file avoids URI fragment ambiguity. If you chose to reuse ID values across non-like-named fields, consider only using URI fragments in href fields that point to resource assemblies.

When resolving a URI fragment for an ID that exists more than once in the file, the FedRAMP PMO encourages tool developers to have their tools evaluate whether an ID points to a resource first.

2.5. Handling of OSCAL Data Types

OSCAL fields and flags have data types assigned to them. NIST provides important information about these data types here:
https://pages.nist.gov/OSCAL/documentation/schema/datatypes/

The following sections describe special handling considerations for data types that directly impact FedRAMP content in OSCAL.

2.5.1. Date and Time in OSCAL Files

Except where noted, all dates and times in the OSCAL-based content must be in an OSCAL dateTime-with-timezone format as documented here:

https://pages.nist.gov/OSCAL/documentation/schema/datatypes/#datetime-with-timezone

This means all dates and times must be represented in the OSCAL file using following format, unless otherwise noted:

`"Y-m-d\TH:i:s.uP"` (See HERE for formatting codes.)

For example, a publication date of 5:30 pm EST, January 10, 2020 must appear as 2020-01-10T17:30:00.00-05:00

This includes:

- Numeric Year: Four-digits
- A dash
- Numeric Month: Two-digit, zero-padded
- A dash
- Numeric Day: Two digit, zero padded
- The capital letter “T” (Do not use lower case)
- Hour: Two digit, zero-padded, 24-hour clock (Use 18 for 6:00 pm)
- A colon
- Minute: Two digit, zero-padded
- A colon
- Seconds: Two digit, zero-padded
- A decimal point
- Fractions of a second: two or three digits, zero padded

Followed by either:

- A capital letter Z to indicate the time is expressed in Coordinated Universal Time (UTC)

OR:

- A plus or minus representing the offset from UTC
- Hour Offset: Difference from UTC, two-digit, padded
- A colon
- Minutes Offset: Difference from UTC, two-digit, padded

This is only for *storing* dates in the OSCAL file. NIST syntax verification tools will generate an error if this format is not found.

Tool developers are encouraged to *present* dates as they have historically appeared in FedRAMP documents. In other words, tools should convert "`2020-03-04T00:00:00.00-05:00`" to "`March 4, 2020`" when presenting the publication date to the user.

Please use the appropriate UTC offset in your region. If you are storing a date and padding the time with zeros, you may also pad the UTC offset with zeros.

2.5.2. Working With href Flags

Several OSCAL fields contain `href` flags. All OSCAL-based `href` flags are URIs formatted according to section 4.1 of RFC3986. When assembling or processing an OSCAL-based FedRAMP file, please consider the following:

Absolute Paths: When using an absolute path within a FedRAMP OSCAL file, the path must be publicly accessible from any location on the Internet, to ensure agency and FedRAMP reviewers can reach the information.

Tool developers are encouraged to validate paths before storing them in OSCAL files and raise issues to users if paths are not reachable.

Relative Paths: All relative paths are assumed to be based on the location of the OSCAL file, unless tools are explicit as to other handling. Sensitive external documents should travel with the OSCAL file and be linked using a relative path.

Internal Locations: These URI fragments appear as just a hashtag (#) followed by a name, such as `#att-diagram-1`. The notation points to a location internal to the document itself and is most commonly used in a FedRAMP OSCAL file as a pointer to `resource` assemblies.

If only a fragment is present, the OSCAL tools must strip the hashtag (#) and treat the remaining string as an ID internal to the OSCAL file itself. For example, the following OSCAL content contains an `href` flag with a URI fragment:

URI Fragment Example

```
    <system-characteristics>
       <authorization-boundary>
          <diagram id="dia-authorization-boundary-1">
             <link href="#att-diagram-boundary-1"/>
             <caption>Authorization Boundary Diagram</caption>
          </diagram>
       </authorization-boundary>
    </system-characteristics>
```

When a tool processes the above example, it should look inside the document for a field or assembly with an ID of "`att-diagram-boundary-1`". This can be accomplished with the following XPath query: `//*[@id="att-diagram-boundary-1"]`

If this is found to point to a resource assembly, see the *Attachments and Embedded Content in OSCAL Files* section for additional handling.

The name of the field or assembly referenced by the above URI fragment can be determined using the following XPath 2.0 query:
`//*[@id="att-diagram-boundary-1"]/name()`

To express this in XPath 1.0, you must use the following:

```
name(//*[@id="att-diagram-boundary-1"])
```

The above query will return "`resource`", if the ID points to a `resource` assembly.

2.5.3. Markup-line and Markup-multiline Fields in OSCAL

As with most machine-readable formats, most of OSCAL's fields are intended to capture short, discrete pieces of information; however, sometimes users require content to be formatted using features such bold, underline, and italics.

For markup-line and markup-multi-line, a subset of HTML is used to format XML-based OSCAL files, while Markdown is used to format JSON-based OSCAL files.

OSCAL provides two types of fields for this purpose:

- **markup-line**: Allows some formatting within a single line of text.
- **Markup-multiline**: Allows all the markup-line formatting, plus allows multiple lines, ordered/ unordered lists, and tables.

A subset of HTML is used to format XML-based OSCAL files, while Markdown is used to format JSON-based OSCAL files.

NIST has implemented only a subset of formatting tags from these standards. This is to ensure formatted content converts between XML and JSON consistently.

Both *markup-line* and *markup-multiline* support:

- emphasis and important text
- inline code and quoted text
- sub/super-script
- images and links

Markup-multiline also supports:

- Paragraphs
- Headings (Levels 1-6)
- Preformatted text
- Ordered and Unordered Lists
- Tables

For a complete list of markup-line and markup-multiline features, please visit:
https://pages.nist.gov/OSCAL/documentation/schema/datatypes/#markup-data-types

2.5.4. Working with Markup-multiline Content

No special handling is required to work with markup-multiline content in JSON, which is based on Markdown syntax; however, XML-based markup-multiline fields require all content to be enclosed in `<p>, <h1> - <h6>, <ol>, <ul>, <pre>`, or `<table>` tags. You may mix several of these within the field.

The following examples offer correct and incorrect representations of markup-multiline content.

The example below is a common misuse of markup-multiline. The description contains text, but the text is not enclosed in one of the required tags. This will produce an error when checked with the OSCAL schema.

Incorrect Markup-multiline Representation

```
    <system-characteristics
       <!-- cut -->
       <description>The xyz system performs ...</description>
    </system-characteristics>
```

The simplest way to correct the error is to enclose the text in <p></p> tags, within the `description` field.

Correct Markup-multiline Representation

```
    <system-characteristics
       <!-- cut -->
       <description><p>The xyz system performs ... </p></description>
    </system-characteristics>
```

The example below demonstrates a correct use of markup-multiline in XML. Please note, the inclusion of a `<p />` tag on a line by itself inserts an empty paragraph. Within XML and HTML, this is treated as a shortcut, and is interpreted as "`<p></p>`"

Correct Markup-multiline Representation

```
    <system-characteristics
       <!-- cut -->
       <description>
          <p>The <b>xyz system</b> performs ... </p>
          <p>The xyz system further supports ... as follows:</p>
          <table>
             <tr>
                <td>Cell A1</td>
                <td>Cell B1</td>
             </tr>
             <tr>
                <td>Cell A2</td>
                <td>Cell B2</td>
             </tr>
          </table>
          <h1>Big Header</h1>
          <p>More detail</p>
          <p><img alt="alt text" src="url" title="title text"/></p>
       </description>
    </system-characteristics>
```

Please also note, all content is enclosed in one of the supported high-level tags (`<p>, <h1> - <h6>, <ol>, <ul>, <pre>`, or `<table>`).

For more information, please visit:
https://pages.nist.gov/OSCAL/documentation/schema/datatypes/#markup-data-types

2.5.5. Special Characters in OSCAL

While OSCAL itself does not directly impose special character handling requirements, XML and JSON do. Characters, such as ampersand (&), greater than (>), less than (<), and quotes (") require special treatment in OSCAL files, depending on the format. For a complete list of special characters and the appropriate treatment for each format, please visit:

https://pages.nist.gov/OSCAL/documentation/schema/datatypes/#specialized-character-mapping

2.6. Citations, Attachments, and Embedded Content in OSCAL Files

All citations, attachments, and embedded content are handled in the `back-matter` section of the OSCAL file as a `resource` assembly. This includes logos, diagrams, policies, procedures, plans, and interconnection security agreements (ICAs). Each `resource` may be referenced from anywhere in the OSCAL file, using its citation ID or resource ID. FedRAMP prefers Base64-encoding for diagrams, such as those related to the authorization boundary, network architecture, and data flows.

```
    <back-matter>
       <!-- citation -->
       <resource id="csp-logo">
          <title>Attachment or Document Title</title>
          <desc>CSP Logo</desc>
          <rlink href="./logo.png" media-type="image/png" />
          <rlink href="./logo.jpg" media-type="image/jpeg" />
          <base64>00000000</base64>
       </resource>
    </back-matter>
```

The `media-type` flag should be an Internet Assigned Numbers Authority (IANA) Media Type when one exists.

2.6.1. Resource Usage Concept

OSCAL is designed so that all attachments, external links, and embedded content may be defined and attached once within the `back-matter` portion of the file. These resources should then be linked using a URI fragment where referenced within the OSCAL file. This enables a tool to update the `back-matter` content once and have the change reflected everywhere it is referenced.

For example, a policy document that satisfies several control families is attached in the back-matter, with an ID of "att-policy-1". Each control covered by that policy, links to the policy using a URI fragment as follows:

```
<link href="#att-policy-1" rel="policy" />
```

2.6.2. General Handling of Multiple/Mixed Links and Base64 Content

Resources may be expressed as:

- one or more links to external files (`rlink`);
- `one or more base64`-encoded data within the OSCAL file;
- any combination of the above; or
- neither.

OSCAL allows multiple rlink fields for the same resource, which are intended to provide options such as links to the same content in different formats, or low- and high-resolution versions of the same graphic.

This may also be used to capture an absolute path to the authoritative location of a file, and a relative path to a local copy of the same file. This is useful when sharing the OSCAL file with reviewers who may not have access to the authoritative location, while preserving the link to that location.

For now, NIST has left multiple/mixed links and base64 content as a topic for developers to address as they see fit. As such, there is currently **no** OSCAL rule which governs how a tool should handle a situation where both the rlink and base64 fields exist, or where there is more than one of either field for a single resource.

2.6.3. FedRAMP's Handling of Multiple/Mixed Links and Base64 Content

FedRAMP tools will always give preference to base64-encoded content. If the `base64` field is present with valid content, FedRAMP tools will ignore all `rlink` fields.

If the `base64` field is missing or invalid, FedRAMP tools will try the first `rlink` field. If that works, all remaining `rlinks` will be ignored. If the first one does not work, FedRAMP tools will try the second link, then the third. It will keep going until either a valid rlink is found, or no rlink fields remain.

XML tools typically respect the sequence of fields in a file. Unfortunately, many JSON tools do not present date in the same sequence present in the file. Please consider prompting the user to select the appropriate field when more than one valid version is available and automated tools are unable to determine the best choice.

Cloud Service Providers (CSPs) should take care when using `rlink` fields to ensure paths are relative to the OSCAL file's location itself, and the referenced items are delivered to FedRAMP with the OSCAL file.

The `resource` assembly could be used with neither `rlink` nor `base64` embedded content. If citing a document that may be attached or embedded later, this approach may be used as a placeholder, or for referencing something in the physical world that has no electronic presence.

Using the `resource` assembly with FedRAMP's additional properties (publication, and version) enables a CSP to cite a document by title, version, and date in one place, then reference it where needed throughout the OSCAL file. This way, the title, version and date can be updated in one place later, and will appear updated everywhere it is referenced.

3. FEDRAMP EXTENSIONS, CONFORMITY TAGS, DEFINED IDENTIFIERS, AND ACCEPTED VALUES

A summary of the FedRAMP extensions, conformity tags, defined identifiers, and accepted values appears in the FedRAMP OSCAL Registry.

NIST designed the core OSCAL syntax to meet model cybersecurity information that is common to any organization and compliance framework. They recognized that each framework and organization may have unique needs. Instead of trying to provide a language that meets each of those unique needs, NIST gave organizations the ability to tailor OSCAL to address specific needs.

FedRAMP has tailored OSCAL by specifying:

- **Extensions**: allow FedRAMP's OSCAL-based content to capture information that is not available in the core OSCAL syntax.
- **Conformity Tags**: This is a special extension that allows FedRAMP processing tools to find required information without human intervention.
- **Defined Identifiers**: FedRAMP defines several unique ID values that must be used in the ID flag of certain OSCAL fields and assemblies.
- **Accepted Values**: For many fields, FedRAMP specifies a case-sensitive set of values. Only these values are recognized by FedRAMP processing tools.

3.1.1. FedRAMP Extensions

All FedRAMP extensions include a namespace (`ns`) flag set to `"https://fedramp.gov/ns/oscal"`.

There are several pieces of information required in FedRAMP templates that cannot be modeled using OSCAL's core syntax. NIST wanted to limit the core OSCAL syntax to those elements that are universal across most cybersecurity frameworks. They designed OSCAL to be extended where unique needs existed.

NIST allows organizations to extend OSCAL anyplace `prop` fields, `part` assemblies, or `annotation` assemblies exist in the core syntax. (Please note, there are currently no `part` assemblies in the SSP, SAP, SAR, or POA&M.) There are two fundamental requirements for extending OSCAL:

- The organization must establish a unique namespace (ns) identifier, such as (`ns="http://domain.tld/ns/oscal"`), and use it to consistently tag all `prop`, `part`, and `annotation` extensions from that organization.
- The organization is responsible for defining, managing, and communicating all names (`name="scan-type"`) defined and tagged with the above name space identifier.

NIST's core OSCAL `prop` fields and `annotation` assemblies have no `ns` flag. If an `ns` flag is present, it is an organization-defined extension. This allows each industry standards body or organization to create their own extensions in their own name space without concern for overlapping names.

The above approach ensures two different organizations can create their own extensions without concern for reusing the same name values. At some point in the future, NIST may provide a registry for organizational extensions. For now, FedRAMP is publishing its own registry to document its extensions.

All FedRAMP extensions include a namespace (`ns`) flag set to " `https://fedramp.gov/ns/oscal`".

For example, if the core OSCAL syntax has a status field, but both FedRAMP and the payment card industry (PCI) require their own framework-specific `status` field, each may define an extension with the `name="status"`, and assign their own `ns` flag. This results in three possible status fields as follows:

NIST OSCAL User Representation

```
<prop name="status">There is no @ns, so this is core OSCAL syntax</prop>
```

XPath Query

```
//prop[@name="user"][not(@ns)]
```

When searching an OSCAL file for a `prop` or `annotation` extensions that is part of the core OSCAL syntax, developers must filter out any with an `ns` flag using the syntax above.

FedRAMP User Representation

```
<prop name="status" ns="https://fedramp.gov/ns/oscal">FedRAMP Status</prop>
```

XPath Query

```
//prop[@name="status"][@ns="https://fedramp.gov/ns/oscal"]
```

(Possible) PCI User Representation

```
<prop name="user" ns="https://pcisecuritystandards.org/ns/oscal">PCI
User</prop>
```

XPath Query

```
//prop[@name="user"][@ns="https://pcisecuritystandards.org/ns/oscal"]
```

* This is an example, and is not intended to represent an actual PCI extension.

Tool developers must always refer to extensions using **both** the `name` and `ns` flags as a pair.

All FedRAMP extensions will appear as either:

```
<prop name="____" ns="https://fedramp.gov/ns/oscal">Value</prop>
```

or:

```
<annotation name="____" ns="https://fedramp.gov/ns/oscal" value="Value">
   <remarks><p>A possible remark about the value</p></remarks>
</annotation>
```

NOTE: The catalog and profile OSCAL models also allow the `part` assembly to be used for extensions. This is not currently the case for the OSCAL SSP, SAP, SAR, or POA&M.

FedRAMP extensions are cited in relevant portions of this document and summarized in the FedRAMP OSCAL Registry.

3.1.2. FedRAMP Conformity Tagging

Some FedRAMP-required content is easily found by processing tools using the core OSCAL syntax; however, OSCAL's versatility makes other required FedRAMP content difficult to identify or ambiguous.

FedRAMP Conformity Tagging is a special extension that allows FedRAMP processing tools to find required information without human intervention, where the core OSCAL syntax alone is not sufficient for validation.

For example, a contingency plan is typically provided in OSCAL as a `resource` attachment in the `back-matter` portion of an OSCAL file. The attachment would include a conformity tag with a value of "`contingency-plan`". This ensures FedRAMP tools know which attachment is the CSP's contingency plan.

Unlike defined identifiers (below), FedRAMP conformity tags are designed to co-exist with conformity tags from other organizations and compliance frameworks.

FedRAMP Conformity Tags are cited in relevant portions of this document and summarized in the FedRAMP OSCAL Registry.

3.1.3. OSCAL and FedRAMP-Defined Identifiers

In limited portions of the OSCAL syntax, such as roles, NIST provides typical *case-sensitive* identifiers common to most OSCAL applications.

FedRAMP carefully expands this list for cloud-base systems, and NIST is evaluating the FedRAMP additions for inclusion in core OSCAL. As of this publication, NIST views all identifiers as draft. As such, they are subject to change until OSCAL 1.0 is released later in calendar year 2020.

For example, every system must have a system owner, and typically has an Information System Security Officer (ISSO). NIST defines specific identifiers for these roles, which enables tools to easily identify the individuals associated with these roles.

FedRAMP defined identifiers are cited in relevant portions of this document and summarized in the FedRAMP OSCAL Registry.

3.1.4. OSCAL and FedRAMP Accepted Values

To facilitate consistent processing, the value for some fields is limited to a list of *case-sensitive* acceptable values. For some fields, OSCAL defines acceptable values, which are enforced by OSCAL-based syntax validation mechanisms.

There are additional fields, where OSCAL syntax validation mechanisms will accept any value, but FedRAMP provides a limited list of *case-sensitive* acceptable values. Where defined, only these values are recognized by FedRAMP processing tools.

For example, every control requires an implementation status. FedRAMP only accepts one of five possible responses for this status, which must be provided using one of the specified choices.

FedRAMP accepted values are cited in relevant portions of this document and summarized in the FedRAMP OSCAL Registry.

4. EXPRESSING COMMON FEDRAMP TEMPLATE ELEMENTS IN OSCAL

While each FedRAMP template has a unique purpose, they share common information elements, such as title and publication date. These common elements are expressed using the same OSCAL syntax for the SSP, SAP, SAR, and POA&M. This section provides OSCAL syntax for these common elements, including:

- Document Basics:
  - Document Title
  - Document Publication Date
  - Document Version
  - Document Sensitivity & Ownership Markings
  - Prepared By
  - Prepared For
  - Document Revision History
- OSCAL-Additional Document Basics:
  - Last Modified Date
  - OSCAL Syntax Version
- Logos:
  - FedRAMP Logo
  - CSP Logo
  - Assessor Logo
  - Consulting 3PAO Logo
- Attachments:
  - FedRAMP Acronyms
  - FedRAMP Citations (Laws, Regulations, Standards, and Guidance)

The following pages are formatted for tabloid (11" x 17") paper in landscape orientation.

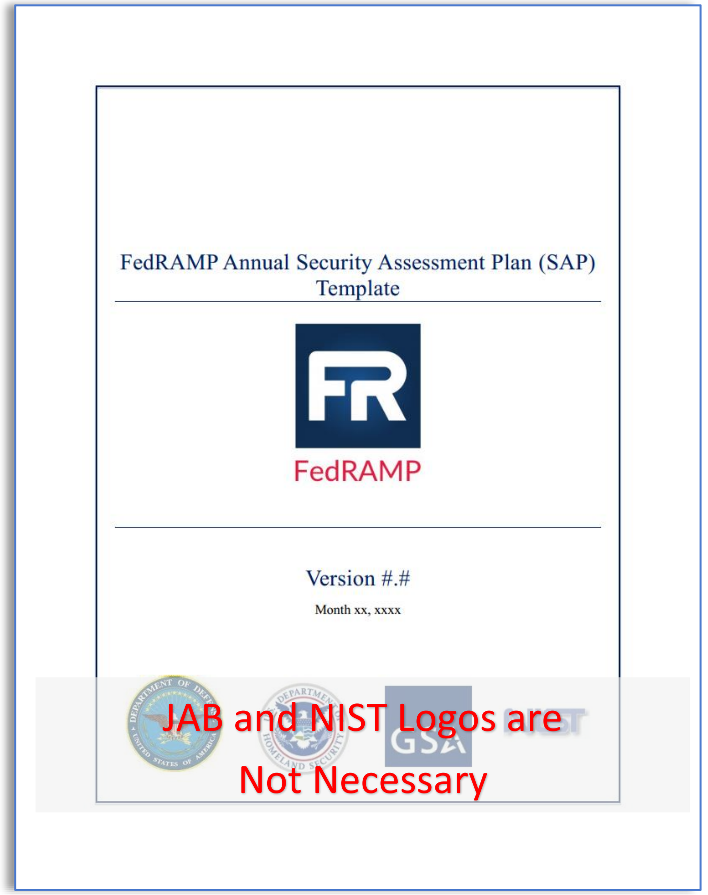
FEDRAMP ___ TEMPLATE

Version #

Version Date

FedRAMP

CONTROLLED UNCLASSIFIED INFORMATION



FedRAMP Annual Security Assessment Plan (SAP) Template

FedRAMP

Version #.#

Month xx, xxxx

JAB and NIST Logos are Not Necessary

The **FedRAMP Logo** is base 64 encoded in the back-matter section of the OSCAL-based FedRAMP Templates, and can be referenced with the following XPath:

```
/*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='fedramp-logo']/../base64
```

4.1. Title Page

Representation

```
  <metadata>
     <title>FedRAMP System Security Plan (SSP)</title>
     <published>2020-06-01T00:00:00.00-04:00</published>
     <last-modified>2020-05-31T00:00:00.00-04:00</last-modified>
     <version>0.0</version>
     <oscal-version>1.0-Milestone3</oscal-version>
  </metadata>

  <!-- OSCAL File Body -->

  <back-matter>
     <resource id="logo-fedramp">
        <desc>FedRAMP Logo</desc>
        <prop name='conformity' ns='https://fedramp.gov/ns/oscal'>fedramp-logo</prop>
        <rlink href="" />
        <base64 filename="FedRAMP_LOGO.png">
        <!-- Base64-encoded Logo Cut -->00000000
        </base64>
     </resource>
  </back-matter>
```

FedRAMP-Conformity Tag

FedRAMP Logo:

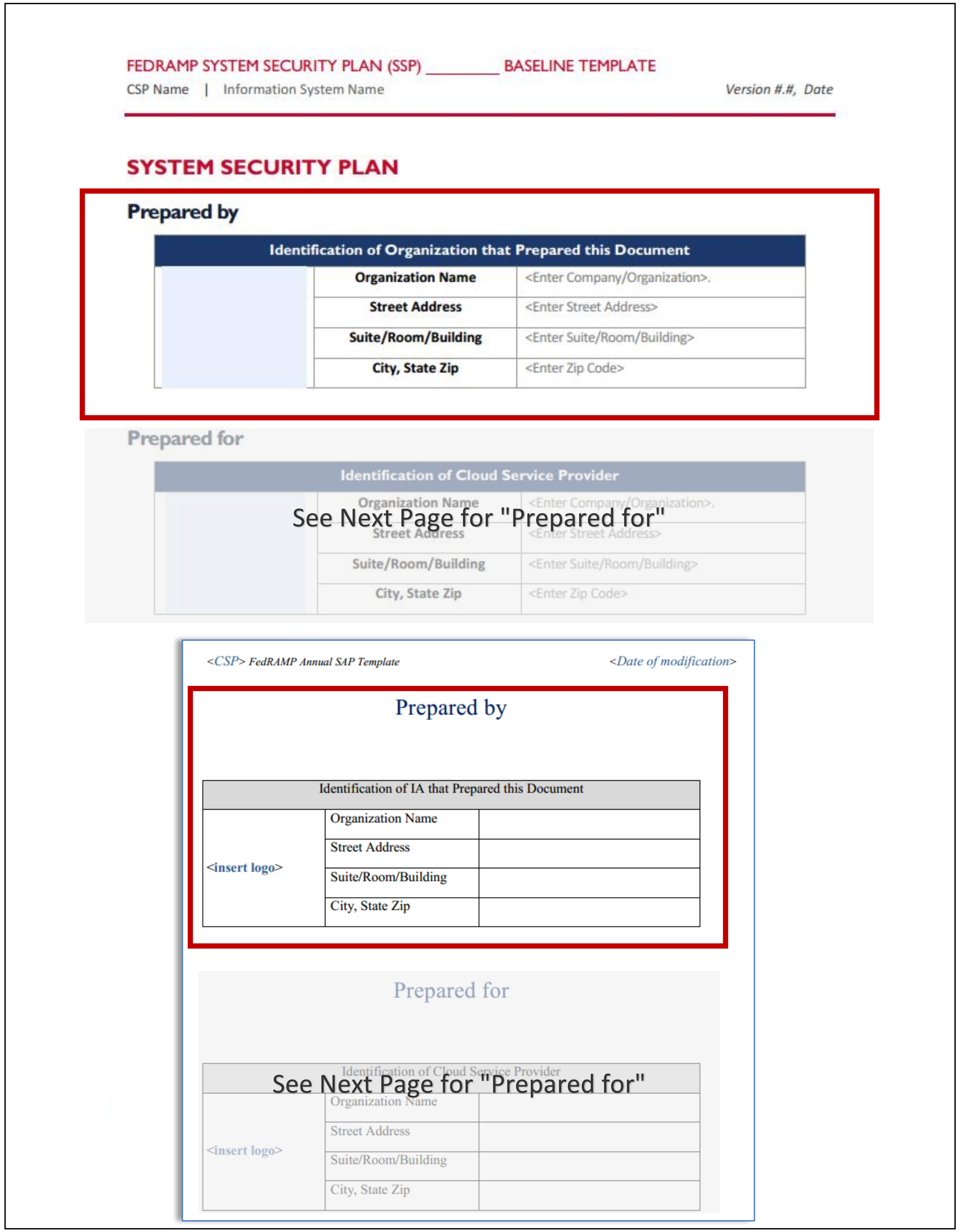
- `fedramp-logo`

XPath Queries

```
Document Title:
  /*/metadata/title

Document Published Version #:
  /*/metadata/version

Document Published Date (will need to convert data for presentation):
  /*/metadata/published

FedRAMP's Logo:
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='fedramp-logo']/../base64

Document Sensitivity Label (may be more than one):
  /*/metadata/prop[@name="marking"]
```

NOTES:

- There may be more than one *Document Sensitivity Label* if needed.
  - Tools should display and/or notify the user of all sensitivity markings.
- The logos on older FedRAMP Templates are not necessary. This includes the NIST Logo, as well as the three Joint Authorization Board (JAB) Agency Logos.



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

CSP Name | Information System Name *Version #.#, Date*

SYSTEM SECURITY PLAN

Prepared by

Identification of Organization that Prepared this Document		
	Organization Name	<Enter Company/Organization>.
	Street Address	<Enter Street Address>
	Suite/Room/Building	<Enter Suite/Room/Building>
	City, State Zip	<Enter Zip Code>

Prepared for

See Next Page for "Prepared for"

Identification of Cloud Service Provider		
	Organization Name	<Enter Company/Organization>.
	Street Address	<Enter Street Address>
	Suite/Room/Building	<Enter Suite/Room/Building>
	City, State Zip	<Enter Zip Code>

<CSP> *FedRAMP Annual SAP Template* *<Date of modification>*

Prepared by

Identification of IA that Prepared this Document		
<insert logo>	Organization Name	
	Street Address	
	Suite/Room/Building	
	City, State Zip	

Prepared for

See Next Page for "Prepared for"

Identification of Cloud Service Provider		
<insert logo>	Organization Name	
	Street Address	
	Suite/Room/Building	
	City, State Zip	

4.2. Prepared By (Third Party)

The FedRAMP SAP and SAR must always indicate the assessing organization using this Prepared By syntax.

Representation

```
<metadata>
   <role id="prepared-by">
      <title>Prepared By</title>
      <desc>The organization that prepared this content.</desc>
   </role>
   <!-- cut: other role assemblies-->
   <!-- cut: location assemblies-->
   <party id="party-1">
      <org>
         <org-name>Name of Consulting Org</org-name>
         <short-name>Acronym/Short Name</short-name>
         <address>
            <!-- address lines cut here for space -->
         </address>
      </org>
   </party>
   <!-- cut: other party assemblies -->
   <responsible-party role-id="prepared-by">
      <party-id>party-2</party-id>
   </responsible-party>
</metadata>

<!-- OSCAL File Body -->

<back-matter>
   <resource id="logo-party-1">
      <desc>Preparer Logo</desc>
      <prop name='conformity' ns='https://fedramp.gov/ns/oscal'>prepared-by-logo</prop>
      <!-- Use rlink and/or base64 -->
      <rlink href="./party-1-logo.png" media-type="image/png" />
      <base64>00000000</base64>
   </resource>
</back-matter>
```

NIST-Defined Identifier

Required Role ID:

- `prepared-by`

FedRAMP-Conformity Tag

Preparer's Logo:

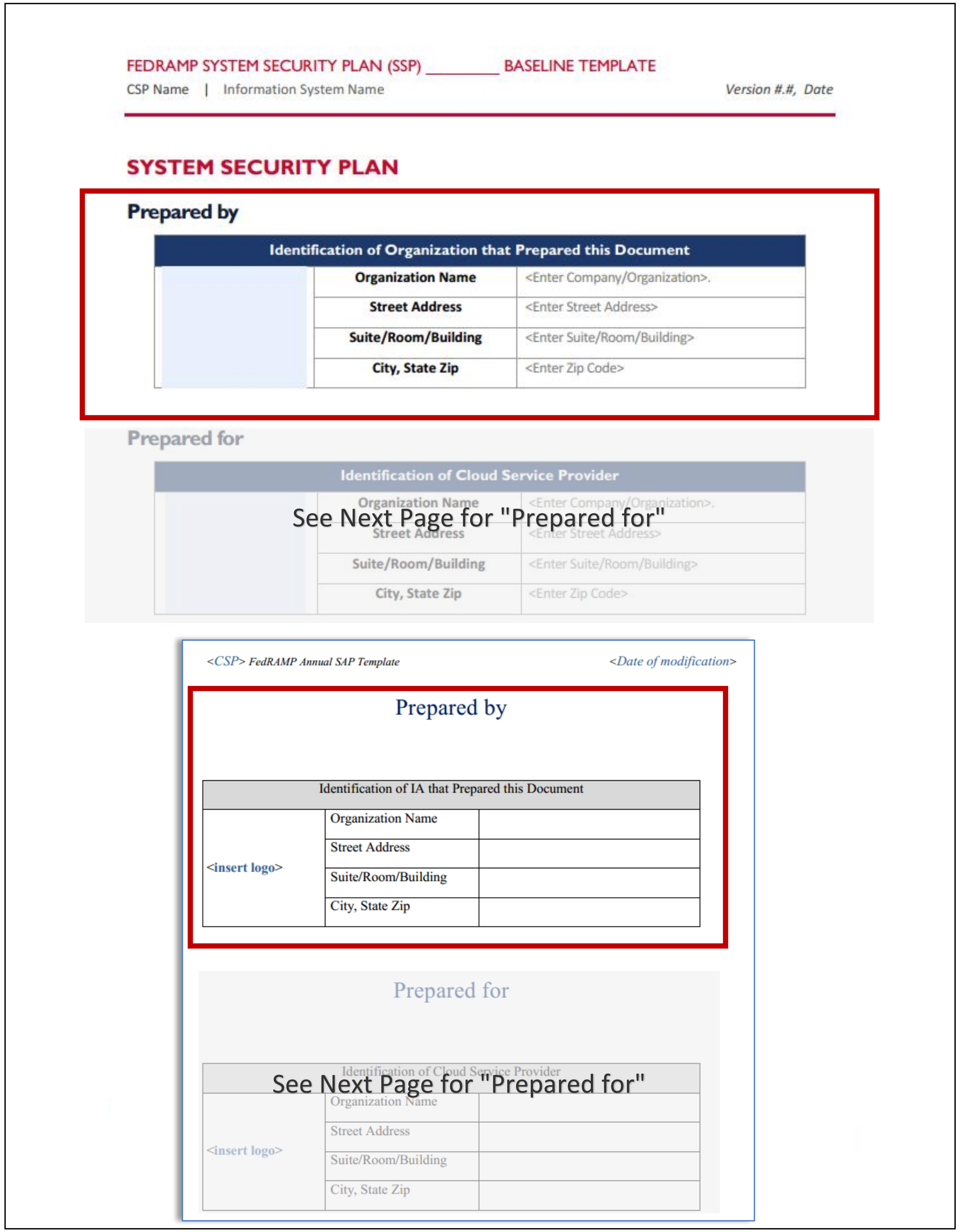
- `prepared-by-logo`

XPath Queries

```
Preparer Organization's Logo:
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='prepared-by-logo']/../base64            OR
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='prepared-by-logo']/../rlink/@href

Preparer Organization's Name and Address:
  /*/metadata/party[@id=[/*/metadata/responsible-party[@role-id='prepared-by']/party-
  id]]/org/org-name

NOTE: Replace "org-name" with "address/addr-line", "address/city", "address/state", or
  "address/zip" as needed. There may be more than one addr-line.
```

NOTES:

- The `responsible-party` assembly connects the role to the party and is required for compliance.
- If the content was prepared by an organization other than the CSP, their logo should also appear in back-matter.



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

CSP Name | Information System Name *Version #.#, Date*

SYSTEM SECURITY PLAN

Prepared by

Identification of Organization that Prepared this Document		
	Organization Name	<Enter Company/Organization>.
	Street Address	<Enter Street Address>
	Suite/Room/Building	<Enter Suite/Room/Building>
	City, State Zip	<Enter Zip Code>

Prepared for

See Next Page for "Prepared for"

Identification of Cloud Service Provider		
	Organization Name	<Enter Company/Organization>.
	Street Address	<Enter Street Address>
	Suite/Room/Building	<Enter Suite/Room/Building>
	City, State Zip	<Enter Zip Code>

<CSP> FedRAMP Annual SAP Template *<Date of modification>*

Prepared by

Identification of IA that Prepared this Document		
<insert logo>	Organization Name	
	Street Address	
	Suite/Room/Building	
	City, State Zip	

Prepared for

See Next Page for "Prepared for"

Identification of Cloud Service Provider		
<insert logo>	Organization Name	
	Street Address	
	Suite/Room/Building	
	City, State Zip	

4.3. Prepared By (CSP Self-Prepared)

This is applicable where the CSP creates or updates its own SSP or POA&M content. The FedRAMP SAP and SAR must never be CSP self-prepared.

Representation

```
<metadata>
   <role id="prepared-by">
      <title>Prepared By</title>
      <desc>The organization that prepared this content.</desc>
   </role>
   <!-- cut: other role assemblies-->
   <!-- cut: location assemblies-->
   <party id="csp">
      <org>
         <org-name>Cloud Service Provider (CSP) Name</org-name>
         <short-name>CSP Acronym/Short Name</short-name>
         <location-id>location-1</location-id>
      </org>
   </party>
   <!-- cut: other party assemblies -->
   <responsible-party role-id="prepared-by">
      <party-id>csp</party-id>
   </responsible-party>
</metadata>

<!-- OSCAL File Body -->

<back-matter>
   <resource id="logo-csp">
      <desc>CSP Logo</desc>
      <prop name='conformity' ns='https://fedramp.gov/ns/oscal'>prepared-by-logo</prop>
      <prop name="conformity" ns="https://fedramp.gov/ns/oscal">csp-logo</prop>
      <!-- Use rlink and/or base64 -->
      <rlink href="./logo.png" media-type="image/png" />
      <base64>00000000</base64>
   </resource>
</back-matter>
```

NIST-Defined Identifier
Required Role ID:

- `prepared-by`

FedRAMP-Conformity Tag
Preparer's Logo:

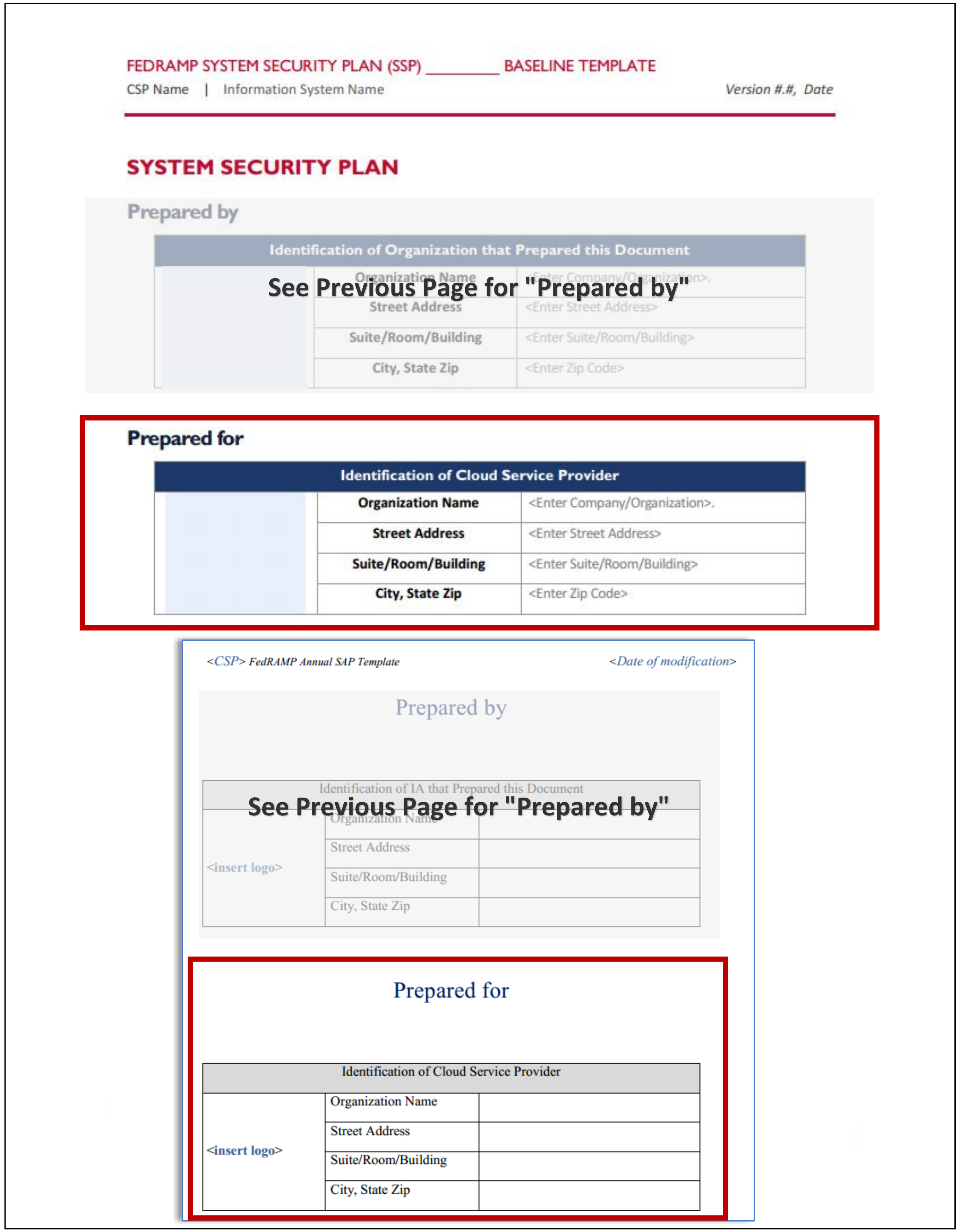
- `prepared-by-logo`

XPath Queries

```
Preparer Organization's Logo:
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='prepared-by-logo']/../base64           OR
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='prepared-by-logo']/../rlink/@href

Preparer Organization's Name and Address:
  /*/metadata/party[@id=[/*/metadata/responsible-party[@role-id='prepared-by']/party-
  id]]/org/org-name

NOTE: Replace "org-name" with "address/addr-line", "address/city", "address/state", or
  "address/zip" as needed. There may be more than one addr-line.
```

NOTES:

- The `responsible-party` assembly connects the role to the party and is required for compliance.
- If the content was prepared by the CSP, the CSP's logo should contain two conformity tags: one to designate the "csp-logo", and another to designate the "prepared-by-logo".



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

CSP Name | Information System Name *Version #.#, Date*

SYSTEM SECURITY PLAN

Prepared by

Identification of Organization that Prepared this Document		
	Organization Name	<Enter Company/Organization>.
	Street Address	<Enter Street Address>
	Suite/Room/Building	<Enter Suite/Room/Building>
	City, State Zip	<Enter Zip Code>

See Previous Page for "Prepared by"

Prepared for

Identification of Cloud Service Provider		
	Organization Name	<Enter Company/Organization>.
	Street Address	<Enter Street Address>
	Suite/Room/Building	<Enter Suite/Room/Building>
	City, State Zip	<Enter Zip Code>

<CSP> FedRAMP Annual SAP Template *<Date of modification>*

Prepared by

Identification of IA that Prepared this Document		
<insert logo>	Organization Name	
	Street Address	
	Suite/Room/Building	
	City, State Zip	

See Previous Page for "Prepared by"

Prepared for

Identification of Cloud Service Provider		
<insert logo>	Organization Name	
	Street Address	
	Suite/Room/Building	
	City, State Zip	

4.4. Prepared For (CSP)

For FedRAMP SSP, SAP, SAR, and POA&M, the "Prepared For" is typically the CSP; however, it may be different if an unforeseen circumstance requires another party to be named. For this reason, "Prepared For" and CSP have separately defined roles.

Representation

```
   <metadata>
      <!-- prop -->
      <role id="prepared-for">
         <title>Prepared For</title>
         <desc>The CSP for FedRAMP SSP, SAP, SAR, and POA&M.</desc>
      </role>
      <role id="cloud-service-provider">
         <title>Cloud Service Provider</title>
         <short-name>CSP</short-name>
      </role>
      <!-- cut: other role assemblies-->
      <!-- cut: location assemblies-->
      <party id="csp">
         <org>
            <org-name>Cloud Service Provider (CSP) Name</org-name>
            <short-name>CSP Acronym/Short Name</short-name>
            <location-id>location-1</location-id>
         </org>
      </party>
      <!-- cut: other party assemblies -->
      <responsible-party role-id="prepared-for">
         <party-id>csp</party-id>
      </responsible-party>
   </metadata>
   <!-- OSCAL File Body -->
   <back-matter>
      <resource id="logo-csp">
         <desc>CSP Logo</desc>
         <prop name="conformity" ns="https://fedramp.gov/ns/oscal">prepared-for-logo</prop>
         <prop name="conformity" ns="https://fedramp.gov/ns/oscal">csp-logo</prop>
         <!-- Use rlink and/or base64 -->
         <rlink href="./logo.png" media-type="image/png" />
         <base64>00000000</base64>
      </resource>
   </back-matter>
```

NIST-Defined Identifiers

Required Role IDs:

- prepared-for
- cloud-service-provider

FedRAMP-Conformity Tags

Prepared For Logo:

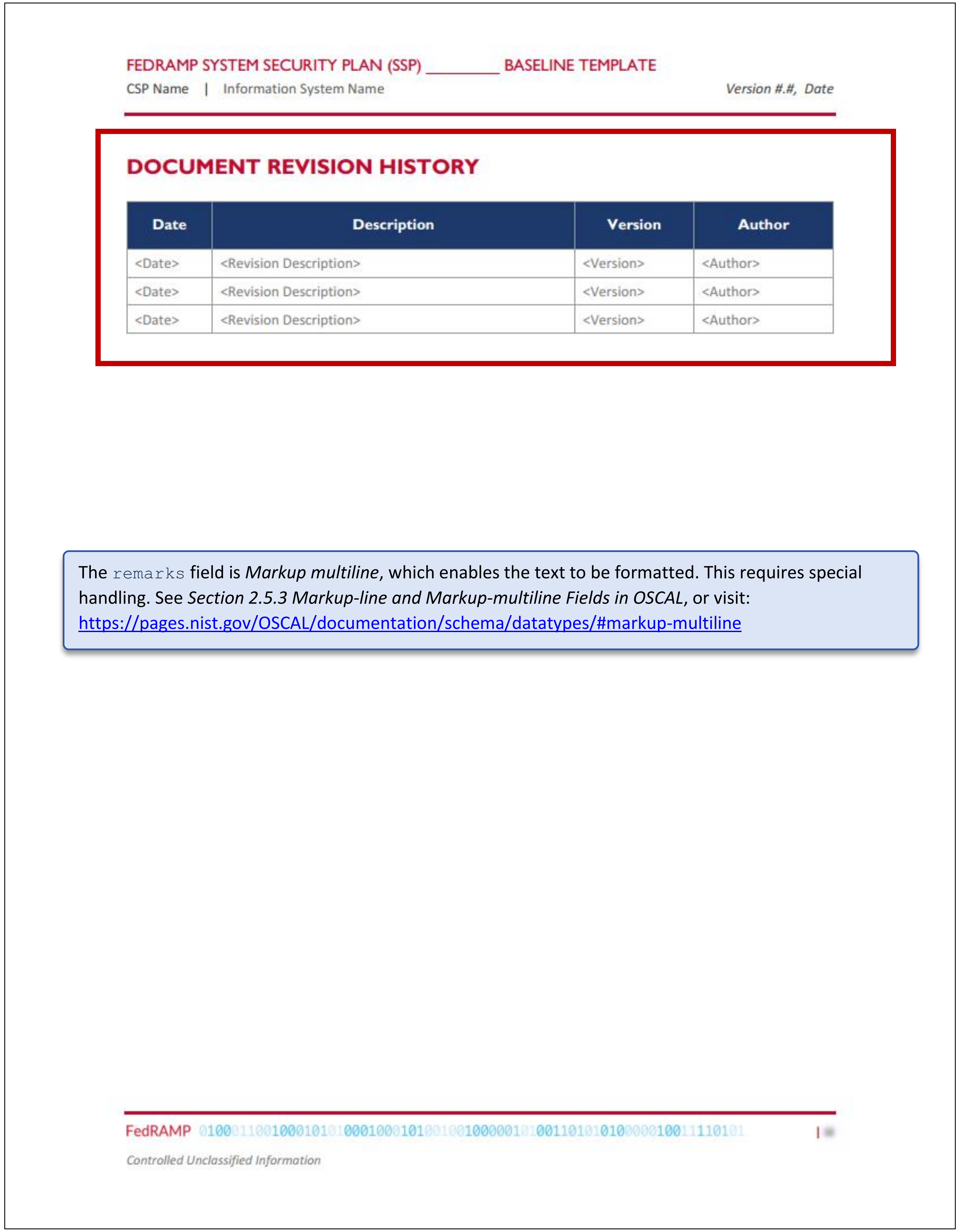
- prepared-for-logo

CSP's Logo:

- csp-logo

XPath Queries

```
CSP’s Logo:
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='csp-logo']/../base64          OR
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='csp-logo']/../rlink/@href

Prepared For (CSP) Details:
  /*/metadata/party[@id=[/*/metadata/responsible-party[@role-id='prepared-for']/party-
  id]]/org/org-name

Prepared For Details:
  /*/metadata/party[@id=[/*/metadata/responsible-party[@role-id='prepared-for']/party-
  id]]/org/org-name

NOTE: Replace "org-name" with "address/addr-line", "address/city", "address/state", or
  "address/zip" as needed. There may be more than one addr-line.
```



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

CSP Name | Information System Name *Version #.#, Date*

DOCUMENT REVISION HISTORY

Date	Description	Version	Author
<Date>	<Revision Description>	<Version>	<Author>
<Date>	<Revision Description>	<Version>	<Author>
<Date>	<Revision Description>	<Version>	<Author>

FedRAMP 0100011001000101010001000101001001000001010011010101000001001110101 |

Controlled Unclassified Information

The `remarks` field is *Markup multiline*, which enables the text to be formatted. This requires special handling. See *Section 2.5.3 Markup-line and Markup-multiline Fields in OSCAL*, or visit: https://pages.nist.gov/OSCAL/documentation/schema/datatypes/#markup-multiline

4.5. Document Revision History

The OSCAL revision history requires one FedRAMP extension to fully meet FedRAMP's revision history requirements.

Representation

```
<metadata>
   <!-- title, published, last-modified, version, oscal-version -->
   <revision-history>
      <revision>
         <published>2019-06-01T00:00:00.00-04:00</published>
         <version>1.0</version>
         <oscal-version>1.0-Milestone3</oscal-version>
         <prop name="party-id" ns="https://fedramp.gov/ns/oscal">consultant</prop>
         <remarks><p>Initial publication.</p></remarks>
      </revision>
      <revision>
         <published>2020-06-01T00:00:00.00-04:00</published>
         <version>2.0</version>
         <oscal-version>1.0-Milestone3</oscal-version>
         <prop name="party-id" ns="https://fedramp.gov/ns/oscal">csp</prop>
         <remarks><p>Updated for annual assessment.</p></remarks>
      </revision>
      <!-- Additional revision assemblies as needed. -->
   </revision-history>
   <!-- doc-id, prop, link, role -->
<metadata>
```

FedRAMP Extension (Author)

prop (`ns="https://fedramp.gov/ns/oscal"`):

- `name="party-id"`

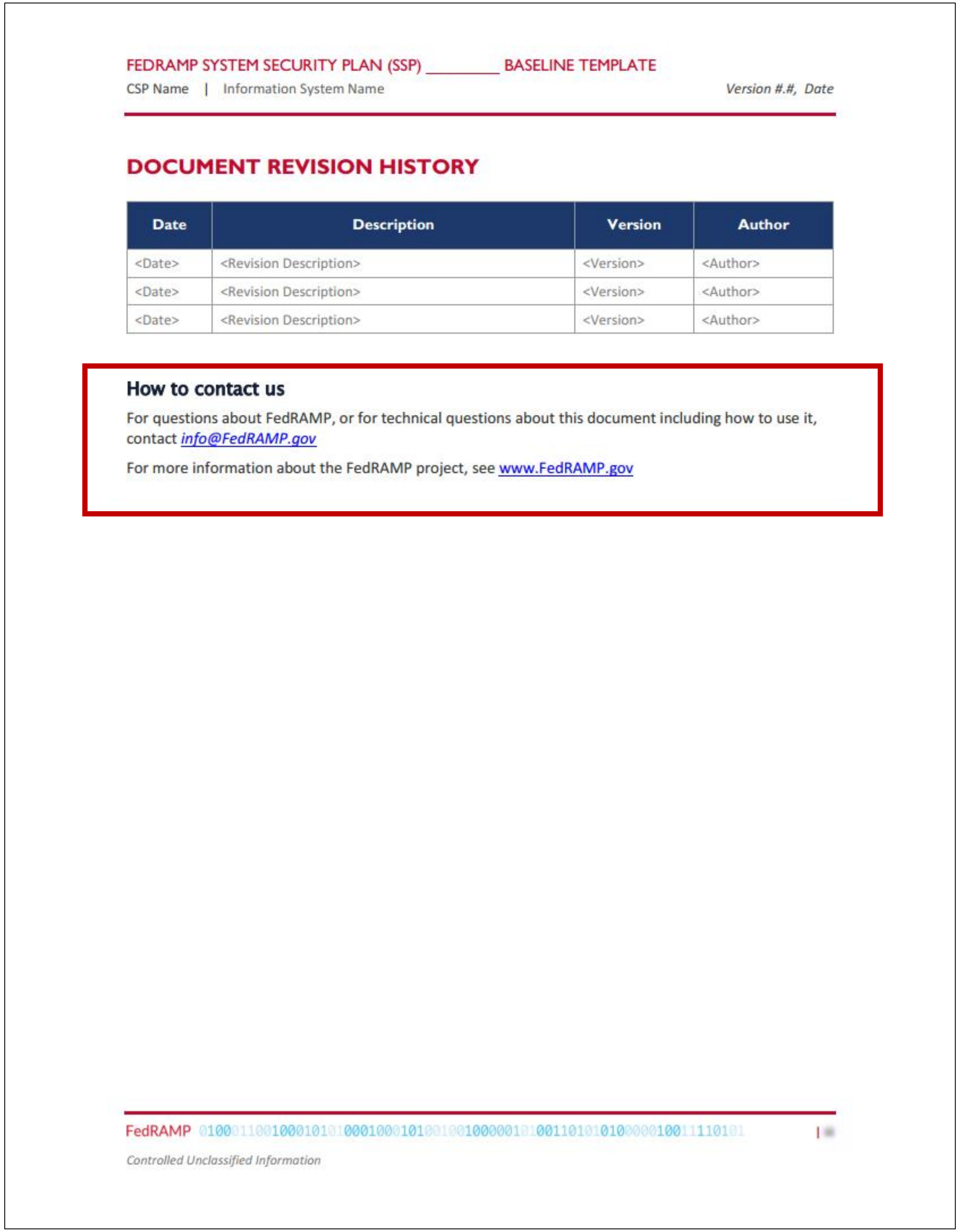
XPath Queries

```
Number of Revision Entries:
  count(/*/metadata/revision-history/revision)

Revision Date for Individual Entry:
  /*/metadata/revision-history/revision[1]/published

Description for Individual Entry:
  /*/metadata/revision-history/revision[1]/remarks/string()

Version for Individual Entry:
  /*/metadata/revision-history/revision[1]/version

Author for Individual Entry:
  /*/metadata/party[@id=/*/metadata/revision-history/revision[1]/prop[@name='party-id']
  [@ns='https://fedramp.gov/ns/oscal']]/org/short-name
```

`Replace "[1]" with "[2]", "[3]", etc.`

NOTES:

- The Revision History's Author field is addressed using a FedRAMP extension, which points to a metadata `party`.
- The published field requires the OSCAL data type, dateTime-with-timezone.
- FedRAMP only requires the publication date, not the time.
  - The time portion may be replaced with all zeros.
  - FedRAMP tools should present only the date, and use a more user-friendly format.



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

CSP Name | Information System Name *Version #.#, Date*

DOCUMENT REVISION HISTORY

Date	Description	Version	Author
<Date>	<Revision Description>	<Version>	<Author>
<Date>	<Revision Description>	<Version>	<Author>
<Date>	<Revision Description>	<Version>	<Author>

How to contact us

For questions about FedRAMP, or for technical questions about this document including how to use it, contact *info@FedRAMP.gov*

For more information about the FedRAMP project, see www.FedRAMP.gov

FedRAMP 0100011001000101010001000101001001000001010011010101000010011110101

Controlled Unclassified Information

4.6. How to Contact Us

The FedRAMP email and web site addresses are part of the organizational content for the FedRAMP PMO.

This content already exists in FedRAMP OSCAL Templates.

Representation

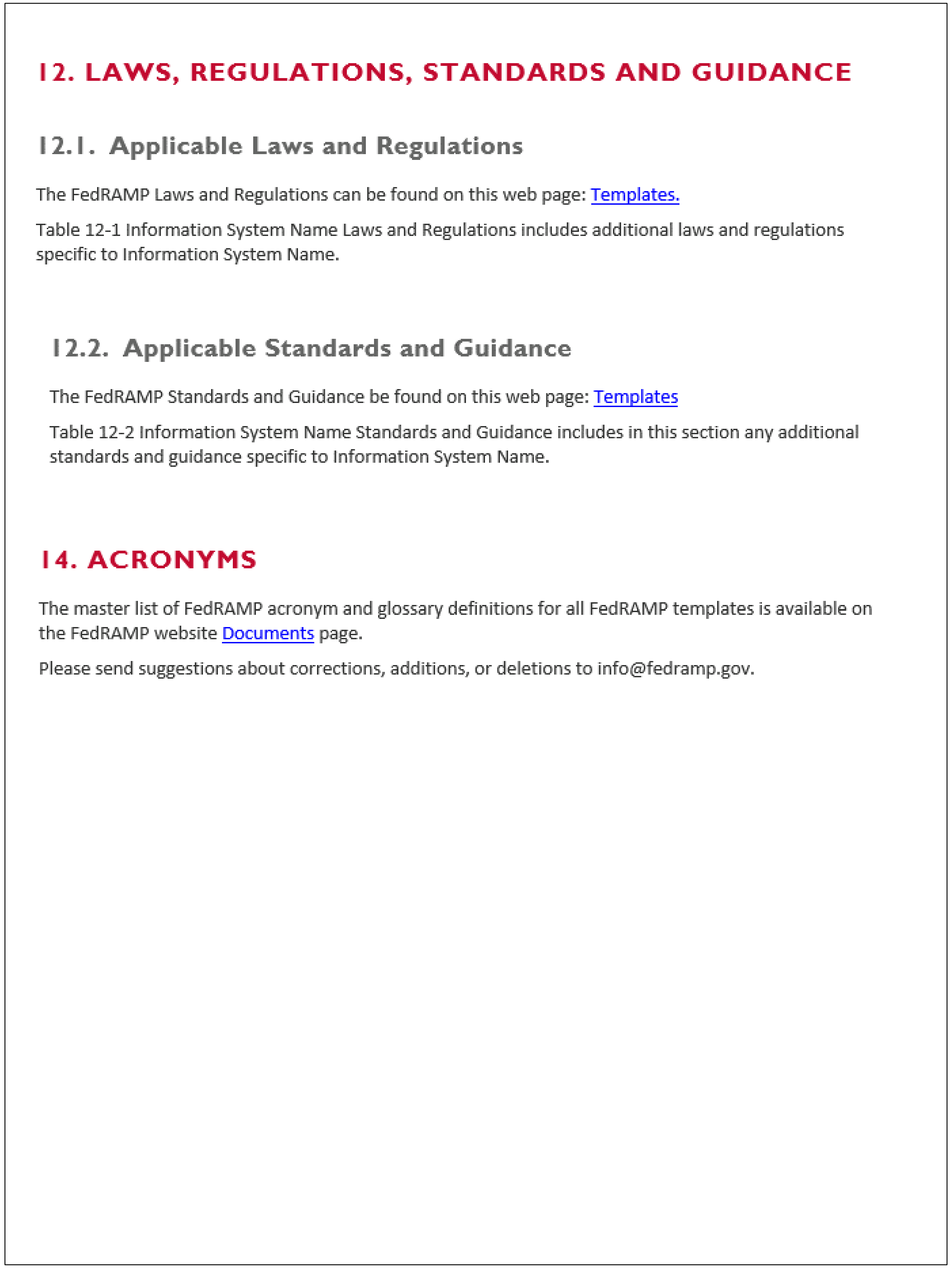
```
...<metadata>
      <!-- role -->
      <party id="fedramp-pmo">
         <org>
            <org-name>Federal Risk and Authorization Management Program: PMO</org-name>
            <short-name>FedRAMP PMO</short-name>
            <address>
               <addr-line>1800 F St. NW</addr-line>
               <city>Washington</city>
               <state>DC</state>
               <postal-code></postal-code>
               <country>us</country>
            </address>
            <email>info@fedramp.gov</email>
            <url>https://fedramp.gov</url>
         </org>
      </party>
      <party id="fedramp-jab">
         <org>
            <org-name>FedRAMP Joint Authorization Board</org-name>
            <short-name>FedRAMP JAB</short-name>
         </org>
      </party>
      <!-- responsible-party -->
...<metadata>
```

FedRAMP-Defined Identifiers:

- `fedramp-pmo`
- `fedramp-jab`

XPath Queries

```
FedRAMP email address:
  /*/metadata/party[@id='fedramp-pmo']/org/email

FedRAMP web site:
  /*/metadata/party[@id='fedramp-pmo']/org/url
```



12. LAWS, REGULATIONS, STANDARDS AND GUIDANCE

12.1. Applicable Laws and Regulations

The FedRAMP Laws and Regulations can be found on this web page: Templates.

Table 12-1 Information System Name Laws and Regulations includes additional laws and regulations specific to Information System Name.

12.2. Applicable Standards and Guidance

The FedRAMP Standards and Guidance be found on this web page: Templates

Table 12-2 Information System Name Standards and Guidance includes in this section any additional standards and guidance specific to Information System Name.

14. ACRONYMS

The master list of FedRAMP acronym and glossary definitions for all FedRAMP templates is available on the FedRAMP website Documents page.

Please send suggestions about corrections, additions, or deletions to info@fedramp.gov.

4.7. FedRAMP Standard Attachments (Acronyms, Laws/Regulations)

The FedRAMP MS Word-based SSP, SAP and SAR templates included links to the FedRAMP Laws and Regulations file, as well as the FedRAMP Acronyms file.

These are already included in the FedRAMP Templates.

Representation

```
<back-matter>
   <resource id="fedramp-1">
      <title>FedRAMP Applicable Laws and Regulations</title>
      <prop name='conformity' ns='https://fedramp.gov/ns/oscal'>fedramp-citations</prop>
      <rlink href="https://www.fedramp.gov/assets/resources/templates/SSP-A12-FedRAMP-Laws-
                  and-Regulations-Template.xlsx" />
   </resource>

   <resource id="fedramp-2">
      <title>FedRAMP Master Acronym and Glossary</title>
      <prop name='conformity' ns='https://fedramp.gov/ns/oscal'>fedramp-acronyms</prop>
      <rlink href=
          "https://fedramp.gov/assets/resources/documents/
          FedRAMP_Master_Acronym_and_Glossary.pdf" />
   </resource>
</back-matter>
```

FedRAMP-Conformity Tags

FedRAMP Standard Citations:

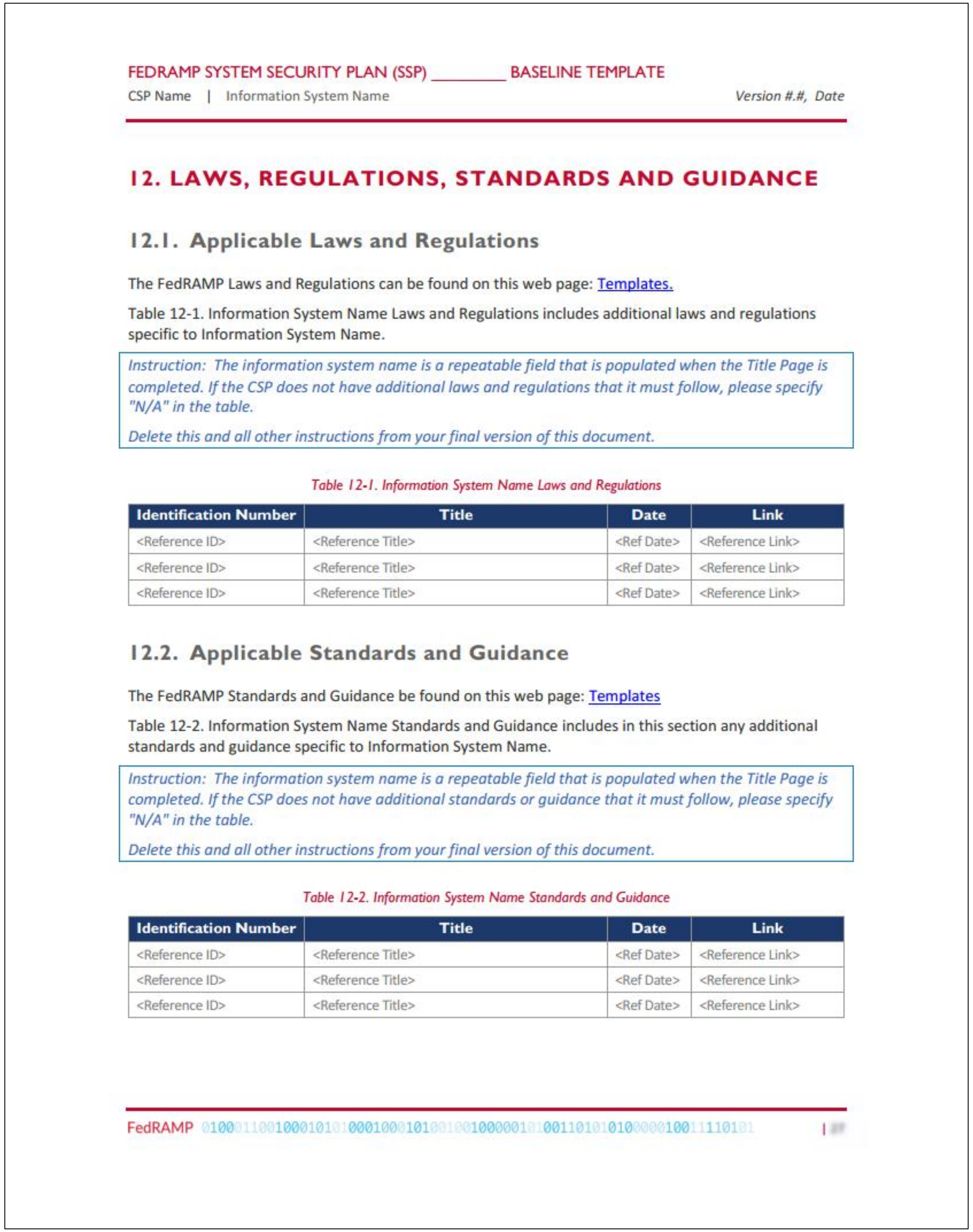
- `fedramp-citations`

FedRAMP Acronyms and Glossary:

- `fedramp-acronyms`

XPath Queries

```
Link to FedRAMP Applicable Laws and Regulations:
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='fedramp-citations']/../rlink/@href

Link to FedRAMP Acronyms and Glossary:
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='fedramp-acronyms']/../rlink/@href
```



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

CSP Name | Information System Name *Version #.#, Date*

12. LAWS, REGULATIONS, STANDARDS AND GUIDANCE

12.1. Applicable Laws and Regulations

The FedRAMP Laws and Regulations can be found on this web page: Templates.

Table 12-1. Information System Name Laws and Regulations includes additional laws and regulations specific to Information System Name.

Instruction: The information system name is a repeatable field that is populated when the Title Page is completed. If the CSP does not have additional laws and regulations that it must follow, please specify "N/A" in the table.

Delete this and all other instructions from your final version of this document.

Table 12-1. Information System Name Laws and Regulations

Identification Number	Title	Date	Link
<Reference ID>	<Reference Title>	<Ref Date>	<Reference Link>
<Reference ID>	<Reference Title>	<Ref Date>	<Reference Link>
<Reference ID>	<Reference Title>	<Ref Date>	<Reference Link>

12.2. Applicable Standards and Guidance

The FedRAMP Standards and Guidance be found on this web page: Templates

Table 12-2. Information System Name Standards and Guidance includes in this section any additional standards and guidance specific to Information System Name.

Instruction: The information system name is a repeatable field that is populated when the Title Page is completed. If the CSP does not have additional standards or guidance that it must follow, please specify "N/A" in the table.

Delete this and all other instructions from your final version of this document.

Table 12-2. Information System Name Standards and Guidance

Identification Number	Title	Date	Link
<Reference ID>	<Reference Title>	<Ref Date>	<Reference Link>
<Reference ID>	<Reference Title>	<Ref Date>	<Reference Link>
<Reference ID>	<Reference Title>	<Ref Date>	<Reference Link>

FedRAMP 0100011001000101010001000101001001000001010011010101000001001110101 | ##

4.8. Additional Laws, Regulations, Standards or Guidance

In addition to the FedRAMP standard laws, regulations, standards, and guidance, the FedRAMP SSP, SAP and SAR templates have a section to cite additional laws and regulations, and another section to cite additional standards and guidance as they relate to the system described by the SSP.

These citations are added as follows:

Representation

```
   <back-matter>
      <resource id="cit-1">
         <title>[SAMPLE]Name or Title of Cited Law</title>
         <prop name="type" ns="https://fedramp.gov/ns/oscal">law</prop>
         <prop name="publication" ns="https://fedramp.gov/ns/oscal">Document Date</prop>
         <prop name="version" ns="https://fedramp.gov/ns/oscal">Document Version</prop>
         <doc-id type="doi">Identification Number</doc-id>
         <rlink href="https://domain.example/path/to/document.pdf" />
      </resource>
      <resource id="cit-2">
         <title>[SAMPLE]Name or Title of Privacy-Related Law Citation</title>
         <prop name="type" ns="https://fedramp.gov/ns/oscal">law</prop>
         <prop name="type" ns="https://fedramp.gov/ns/oscal">pii</prop>
         <prop name="publication" ns="https://fedramp.gov/ns/oscal">Document Date</prop>
         <prop name="version" ns="https://fedramp.gov/ns/oscal">Document Version</prop>
         <doc-id type="doi">Identification Number</doc-id>
         <rlink href="https://domain.example/path/to/document.pdf" />
      </resource>
      <!-- repeat citation assembly for each law, regulation, standard or guidance -->
      <!-- resource -->
   </back-matter>
```

FedRAMP Extensions and Accepted Values

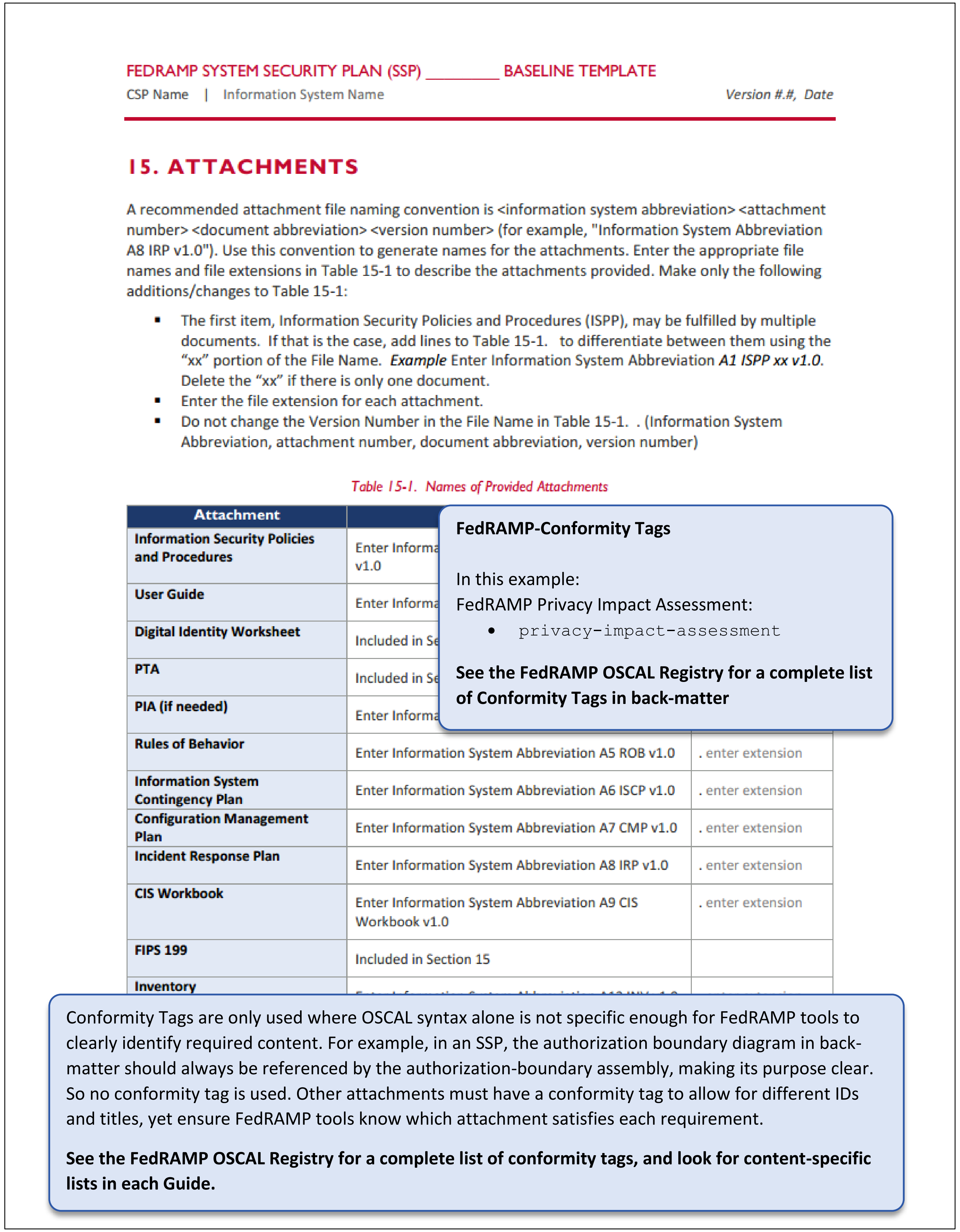
prop (`ns="https://fedramp.gov/ns/oscal"`):

- `name="type"`
  - o **Valid:** `law, regulation, standard, guidance, pii`
- `name="publication"`

XPath Queries

Replace "[1]" with "[2]", "[3]", etc.

```
Number of Laws and Regulations:
  count(/*/back-matter/resource/prop[@name="type"][@ns="https://fedramp.gov/ns/oscal"]
  [(string(.) = "law") or (string(.)="regulation")])

Laws and Regulations - Identification Number:
  (/*/back-matter/resource/prop[@name="type"][@ns="https://fedramp.gov/ns/oscal"]
  [(string(.) = "law") or (string(.)="regulation")])[1]/../doc-id

Laws and Regulations - Title:
  (/*/back-matter/resource/prop[@name="type"][@ns="https://fedramp.gov/ns/oscal"]
  [(string(.) = "law") or (string(.)="regulation")])[1]/../title

Laws and Regulations - Date:
  (/*/back-matter/resource/prop[@name="type"][@ns="https://fedramp.gov/ns/oscal"]
  [(string(.) = "law") or (string(.)="regulation")])[1]/
  ../prop[@name="publication"][@ns="https://fedramp.gov/ns/oscal"]

Laws and Regulations - Link:
  (/*/back-matter/resource/prop[@name="type"][@ns="https://fedramp.gov/ns/oscal"]
  [(string(.) = "law") or (string(.)="regulation")])[1]/../rlink/@href

For Standards and Guidance replace "law" with "standard" and "regulation" with
  "guidance" in the above queries.
```



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

CSP Name | Information System Name *Version #.#, Date*

15. ATTACHMENTS

A recommended attachment file naming convention is <information system abbreviation> <attachment number> <document abbreviation> <version number> (for example, "Information System Abbreviation A8 IRP v1.0"). Use this convention to generate names for the attachments. Enter the appropriate file names and file extensions in Table 15-1 to describe the attachments provided. Make only the following additions/changes to Table 15-1:

- The first item, Information Security Policies and Procedures (ISPP), may be fulfilled by multiple documents. If that is the case, add lines to Table 15-1. to differentiate between them using the "xx" portion of the File Name. *Example* Enter Information System Abbreviation *A1 ISPP xx v1.0*. Delete the "xx" if there is only one document.
- Enter the file extension for each attachment.
- Do not change the Version Number in the File Name in Table 15-1. . (Information System Abbreviation, attachment number, document abbreviation, version number)

Table 15-1. Names of Provided Attachments

Attachment		
Information Security Policies and Procedures	Enter Informa v1.0	
User Guide	Enter Informa	
Digital Identity Worksheet	Included in Se	
PTA	Included in Se	
PIA (if needed)	Enter Informa	
Rules of Behavior	Enter Information System Abbreviation A5 ROB v1.0	. enter extension
Information System Contingency Plan	Enter Information System Abbreviation A6 ISCP v1.0	. enter extension
Configuration Management Plan	Enter Information System Abbreviation A7 CMP v1.0	. enter extension
Incident Response Plan	Enter Information System Abbreviation A8 IRP v1.0	. enter extension
CIS Workbook	Enter Information System Abbreviation A9 CIS Workbook v1.0	. enter extension
FIPS 199	Included in Section 15	
Inventory		

FedRAMP-Conformity Tags

In this example:
FedRAMP Privacy Impact Assessment:

- `privacy-impact-assessment`

See the FedRAMP OSCAL Registry for a complete list of Conformity Tags in back-matter

Conformity Tags are only used where OSCAL syntax alone is not specific enough for FedRAMP tools to clearly identify required content. For example, in an SSP, the authorization boundary diagram in back-matter should always be referenced by the authorization-boundary assembly, making its purpose clear. So no conformity tag is used. Other attachments must have a conformity tag to allow for different IDs and titles, yet ensure FedRAMP tools know which attachment satisfies each requirement.

See the FedRAMP OSCAL Registry for a complete list of conformity tags, and look for content-specific lists in each Guide.

4.9. Attachments and Embedded Content

There are several attachments in a classic FedRAMP MS Word based SSP, SAP, SAR document or Deviation Request (DR) form. Some lend well to machine-readable format, while others do not. Those that are readily modeled in machine-readable format are typically addressed within the OSCAL syntax, while attachments such as policies, procedures, plans, guides, and rules of behavior documents are all treated as attachments in OSCAL as well.

Further, any diagrams or images that normally appear in context, such as the authorization boundary diagram, are attached in the back-matter and referenced from the body of the OSCAL file, as described in Section 2.6 *Citations, Attachments, and Embedded Content in OSCAL Files*. The following table represents attachments and embedded content.

FedRAMP Extensions and Accepted Values

prop (`ns="https://fedramp.gov/ns/oscal"`):

- `name="type"`
  - **`Valid:`** `policy, procedure, guide, pia, rob, plan`
- `name="publication"`
- `name="version"`

Attachment Representation

```
<!-- cut -->
<back-matter>
  <!-- citation -->
  <resource id="att-pia">
    <title>Privacy Impact Assessment</title>
    <desc>Privacy Impact Assessment</desc>
    <prop name="conformity" ns="https://fedramp.gov/ns/oscal">privacy-impact-assessment</prop>
    <prop name="publication" ns="https://fedramp.gov/ns/oscal">Document Date</prop>
    <prop name="version" ns="https://fedramp.gov/ns/oscal">Document Version</prop>
    <!-- Use rlink and/or base64 -->
    <!-- Add rlink with relative path or embed with base64 encoding -->
    <rlink href="./pia.docx" />
    <base64>00000000</base64>
  </resource>
  <resource id="diag-boundary-1">
    <desc>The primary authorization boundary diagram.</desc>
    <!-- Use rlink and/or base64 -->
    <rlink href="./boundary.png" />
    <!-- Add rlink with relative path or embed with base64 encoding -->
    <base64>00000000</base64>
    <remarks><p>Set system-characteristics/authorization-boundary/diagram/link/@href =
              "#diag-boundary-1"</p></remarks>
  </resource>
</back-matter>
```

XPath Queries

```
Replace "policy" with "plan", "rob",
etc. for each attachment type.
```

```
PIA Attachment (Embedded Base64 encoded):
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='privacy-impact-assessment']/.. /base64

PIA Attachment (Relative Link):
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='privacy-impact-assessment']/.. /rlink/@href

Publication Date of PIA:
  /*/back-matter/resource/prop[@name='conformity'][@ns='https://fedramp.gov/ns/oscal']
  [string(.)='privacy-impact-assessment']/../prop[@name="publication"]
  [@ns="https://fedramp.gov/ns/oscal"]
```

APPENDICIES

APPENDIX A. OSCAL-BASED FEDRAMP BASELINES

The system's identified security categorization level governs which FedRAMP baseline applies. This can be checked using the XPath syntax below.

Security Sensitivity Level XPath Query

```
Security Categorization Level:
  /*/system-characteristics/security-sensitivity-level
```

This determines which URL should be entered for the `import-profile` field in an OSCAL-based FedRAMP SSP.

Baseline Representation

```
    <!-- metadata -->
    <!-- This must point to the appropriate FedRAMP Baseline -->
    <import-profile href="https://path/to/FedRAMP_MODERATE-baseline_profile.xml"/>
    <!-- system-characteristics -->
```

FedRAMP validation tools will compare the identified security categorization level to the actual FedRAMP baseline specified in the SSP and raise a warning if a different baseline was used.

High

XML Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/xml/FedRAMP_HIGH-baseline_profile.xml

JSON Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/json/FedRAMP_HIGH-baseline_profile.json

Moderate

XML Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/xml/FedRAMP_MODERATE-baseline_profile.xml

JSON Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/json/FedRAMP_MODERATE-baseline_profile.json

Low

XML Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/xml/FedRAMP_LOW-baseline_profile.xml

JSON Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/json/FedRAMP_LOW-baseline_profile.json

Do not copy and modify the FedRAMP baseline. FedRAMP will use the original, published file for validation, ignoring any modified copies.

If you require a modification to the FedRAMP baselines, such as may be required when directed to do so by an authorizing official, first contact FedRAMP to coordinate the modification, then follow the instructions in Appendix B.

FedRAMP Tailored

FedRAMP Tailored for Low Impact – Software as a Service (LI-SaaS) Appendix B merges SSP, SAP, and SAR information into a single document. The SSP portions of that document may be represented using the same OSCAL conventions described in this document with only a few minor differences.

Specific OSCAL-based FedRAMP Tailored guidance will be published at a later date; however, fully representing Appendix B in OSCAL requires the SSP, SAP, and SAR syntax, used the same way as they are explained for FedRAMP Low, Moderate, and High baselines.

For your convenience, FedRAMP has made the FedRAMP Tailored for LI-SaaS baseline available now in both XML and JSON formats as follows:

Low-Impact SaaS (Tailored)

XML Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/xml/FedRAMP_LI-SaaS-baseline_profile.xml

JSON Version:
https://raw.githubusercontent.com/usnistgov/OSCAL/master/content/fedramp.gov/json/FedRAMP_LI-SaaS-baseline_profile.json

Profile Resolution

Normally, OSCAL is designed to import each layer from the previous layer. While this ensures traceability of selected controls and modified content, it can also be processor intensive. Profile resolution flattens or merges a profile and its imported catalog(s) and profiles into a single OSCAL file using the catalog syntax.

This single file essentially pre-processes the profile -> catalog relationship. This is useful for the FedRAMP baselines given their static nature. Any tool that would normally open an OSCAL-based FedRAMP profile and process it against the NIST SP 800-53 catalog can instead simply use the resolved-profile catalog.

Each FedRAMP XML and JSON baseline profile has a resolved profile catalog in the same location. Where available, these may be used by tools to save processing time.

APPENDIX B. MODIFYING A FEDRAMP BASELINE

OSCAL is designed to allow modification of controls and baselines, while maintaining traceability through each layer of modification. This means you should never copy and modify a FedRAMP baseline.

If you require a change to a FedRAMP baseline, you should first coordinate that change with the FedRAMP JAB or PMO. Assuming FedRAMP agrees with the change, the correct way to implement the change is as follows:

1. **Create a new, blank OSCAL Profile.**
2. **Point to the FedRAMP Baseline**: Point it to the appropriate FedRAMP baseline using an `import` field.
3. **Select the Relevant Controls**: Use the `include` and `exclude` fields to identify the controls to include or exclude from the FedRAMP baseline.
   a. For example, if you need all but one control, you can `include all`, then `exclude` the one.
4. **Specify How Controls Are Organized**: FedRAMP prefers you merge "as-is" using those merge fields. This is relevant when resolving the profile. See the *Profile Resolution* section of this appendix for more information.
5. **Modify the Selected Controls**: Use the `modify` assembly to make modifications to parameters and control definitions.

The next page contains an example profile, which accomplishes the following actions:

- Imports the FedRAMP Moderate baseline
- Includes all controls from that baseline
- Explicitly removes AT-4 from the baseline
- Indicates that if this profile is resolved, the organization of the controls should remain as-is. See the *Profile Resolution* section of this appendix for more information.
- Adds a constraint to the third parameter of AC-1 (ac-1_prm_3), which is more restrictive than the FedRAMP constraint, but changing it from "at least annually" to "at least every six months."
- Removes the additional FedRAMP requirement statement in AU-11 and replaces it with a more restrictive statement, which now requires online retention of audit records for at least 180 days instead of 90 days.

For more information on working with profiles, please visit the NIST OSCAL site at: https://pages.nist.gov/OSCAL

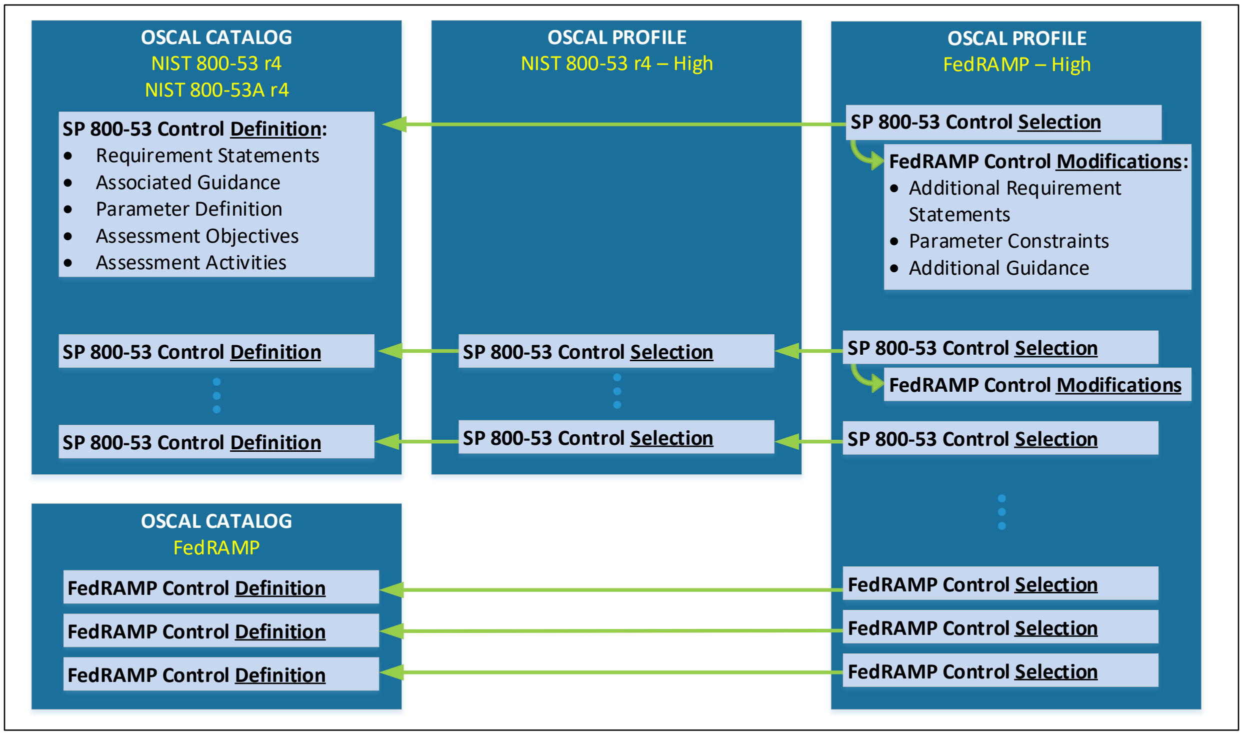
A complete OSCAL profile syntax reference is available here: https://pages.nist.gov/OSCAL/documentation/schema/profile/

Sample Profile to Modify a FedRAMP Baseline

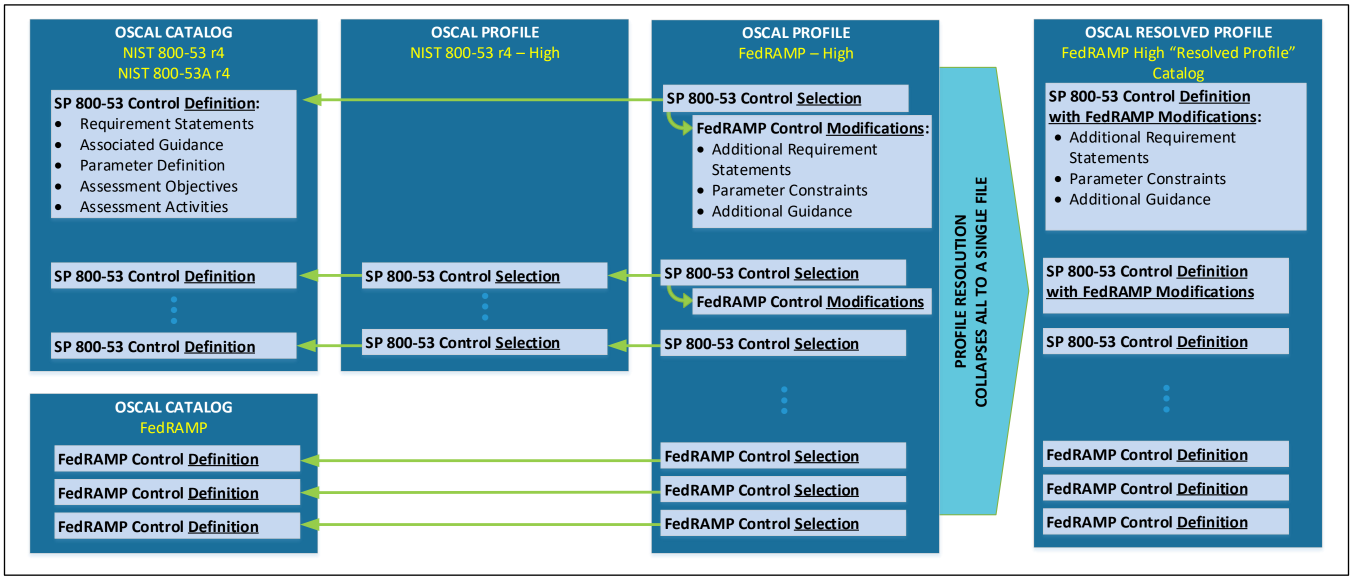
```
<profile xmlns="http://csrc.nist.gov/ns/oscal/1.0"
      id="uuid-xxxx">
      <metadata>
            <title>[XYZ Org] Modification to FedRAMP Moderate
Baseline</title>
            <last-modified>2019-10-01T11:03:27.392-04:00</last-modified>
            <version>1.1</version>
            <oscal-version>1.0.0-milestone2</oscal-version>
      </metadata>
      <import href="https://path/to/FedRAMP_MODERATE-baseline_profile.xml">
            <include>
                  <!-- Include every control in the Moderate baseline -->
                  <all with-child-controls="yes" />
            </include>
            <exclude>
                  <!-- Remove Control AT-4 -->
                  <call control-id="at-4" />
            </exclude>
      </import>
      <merge><as-is>yes</as-is></merge>
      <modify>
            <set param-id="ac-1_prm_3">
                  <!-- Change the constraint from "at least annually" -->
                  <constraint>at least every six months</constraint>
            </set>
            <remove id-ref="au-11_fr" />
            <alter control-id="au-11">
                  <add position="ending">
                        <part id="au-11_fr" name="item">
                              <title>[XYZ Org]Modified Requirement</title>
                              <part id="au-11_fr_smt.1" name="item">
                                    <prop name="label">Requirement:</prop>
                                    <p>The service provider retains audit
records on-line for at 180 days and further preserves audit records off-line
for a period that is in accordance with NARA requirements.</p>
                              </part>
                        </part>
                  </add>
            </alter>
      </modify>
</profile>
```

Profile Resolution

Profiles are intended to identify upstream sources of control definition information and show only the changes to those upstream sources. This enables humans and computers to trace control definition changes back to their source framework.


OSCAL CATALOG
NIST 800-53 r4
NIST 800-53A r4
SP 800-53 Control Definition:
Requirement Statements
Associated Guidance
Parameter Definition
Assessment Objectives
Assessment Activities
SP 800-53 Control Definition
SP 800-53 Control Definition
OSCAL PROFILE
NIST 800-53 r4 – High
SP 800-53 Control Selection
SP 800-53 Control Selection
OSCAL PROFILE
FedRAMP – High
SP 800-53 Control Selection
FedRAMP Control Modifications:
Additional Requirement Statements
Parameter Constraints
Additional Guidance
SP 800-53 Control Selection
FedRAMP Control Modifications
SP 800-53 Control Selection
FedRAMP Control Selection
FedRAMP Control Selection
FedRAMP Control Selection
OSCAL CATALOG
FedRAMP
FedRAMP Control Definition
FedRAMP Control Definition
FedRAMP Control Definition

An organization may prefer to work with a single, complete file containing only the final set of relevant control definitions. OSCAL provides a process for merging this collection of catalogs and profiles into a single catalog. This process is called "profile resolution."


OSCAL CATALOG
NIST 800-53 r4
NIST 800-53A r4
SP 800-53 Control Definition:
Requirement Statements
Associated Guidance
Parameter Definition
Assessment Objectives
Assessment Activities
SP 800-53 Control Definition
SP 800-53 Control Definition
OSCAL CATALOG
FedRAMP
FedRAMP Control Definition
FedRAMP Control Definition
FedRAMP Control Definition
OSCAL PROFILE
NIST 800-53 r4 – High
SP 800-53 Control Selection
SP 800-53 Control Selection
OSCAL PROFILE
FedRAMP – High
SP 800-53 Control Selection
FedRAMP Control Modifications:
Additional Requirement Statements
Parameter Constraints
Additional Guidance
SP 800-53 Control Selection
FedRAMP Control Modifications
SP 800-53 Control Selection
FedRAMP Control Selection
FedRAMP Control Selection
FedRAMP Control Selection
PROFILE RESOLUTION
COLLAPSES ALL TO A SINGLE FILE
OSCAL RESOLVED PROFILE
FedRAMP High "Resolved Profile" Catalog
SP 800-53 Control Definition with FedRAMP Modifications:
Additional Requirement Statements
Parameter Constraints
Additional Guidance
SP 800-53 Control Definition with FedRAMP Modifications
SP 800-53 Control Definition
FedRAMP Control Definition
FedRAMP Control Definition
FedRAMP Control Definition

The `merge` assembly within an OSCAL profile offers a profile creator control over how the final file is organized. To maintain the same organization, simply use the `as-is` field and set it to "yes."

The complete profile syntax is available here:
https://pages.nist.gov/OSCAL/documentation/schema/profile/

APPENDIX C. WORKING WITH ROLES, PEOPLE, AND ORGANIZATIONS

An OSCAL file defines roles, people, and organizations within the metadata as part of three separate assemblies:

- **`role`**: A role ID and role title are required. Other content, such as a short-name, description, or remarks are optional.
- **`party`**: People and organizations are defined next as parties. An organization is any collection of people, and can represent a company, agency, department, or team.
- **`responsible-party`**: Links roles to parties. The same role can have more than one party assigned to it. Also a party can be assigned to more than one role.

Working with Role Identifiers

All roles within the document are defined under the metadata element as follows:

```
<metadata>
   --- cut ---
   <role id="prepared-by">
      <title>Prepared By</title>
      <desc>The organization that prepared this SSP.</desc>
   </role>
   <role id="prepared-for">
      <title>Prepared For</title>
      <desc>The organization for which this SSP was prepared</desc>
   </role>
   --- cut ---
</metadata>
```

To ensure consistent processing, FedRAMP has defined a specific set of roles that must exist with a FedRAMP SSP, SAP, SAR, or POA&M. **Most are pre-populated in the OSCAL-based FedRAMP Templates.** They vary by template. OSCAL-based FedRAMP tools must ensure these roles, titles, and descriptions exist within an OSCAL-based FedRAMP content. Additional roles may be added, provided these roles remain.

NIST-defined and FedRAMP-defined role-identifiers are cited in relevant portions of each guide, and summarized in the FedRAMP OSCAL Registry.

FedRAMP Defined Party Identifiers

To ensure compatibility with multiple compliance frameworks, FedRAMP has depreciated the use of FedRAMP-Defined Party Identifiers.