

This section provides a summary of several important concepts and details that apply to OSCAL-based FedRAMP SSP files.

The [Guide to OSCAL-based FedRAMP Content](#) provides important concepts necessary for working with any OSCAL-based FedRAMP file. Familiarization with those concepts is important to understanding this guide.

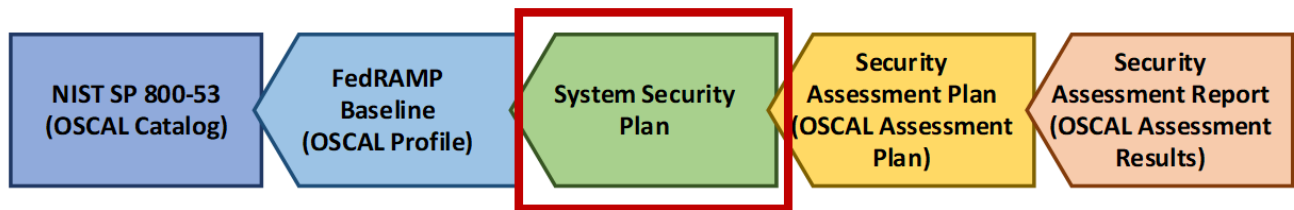
XML and JSON Formats

The examples provided here are in XML; however, FedRAMP accepts XML or JSON formatted OSCAL-based SSP files. NIST offers a utility that provides lossless conversion of OSCAL-compliant files between XML and JSON in either direction.

You may submit your SSP to FedRAMP using either format. If necessary, FedRAMP tools will convert the files for processing.

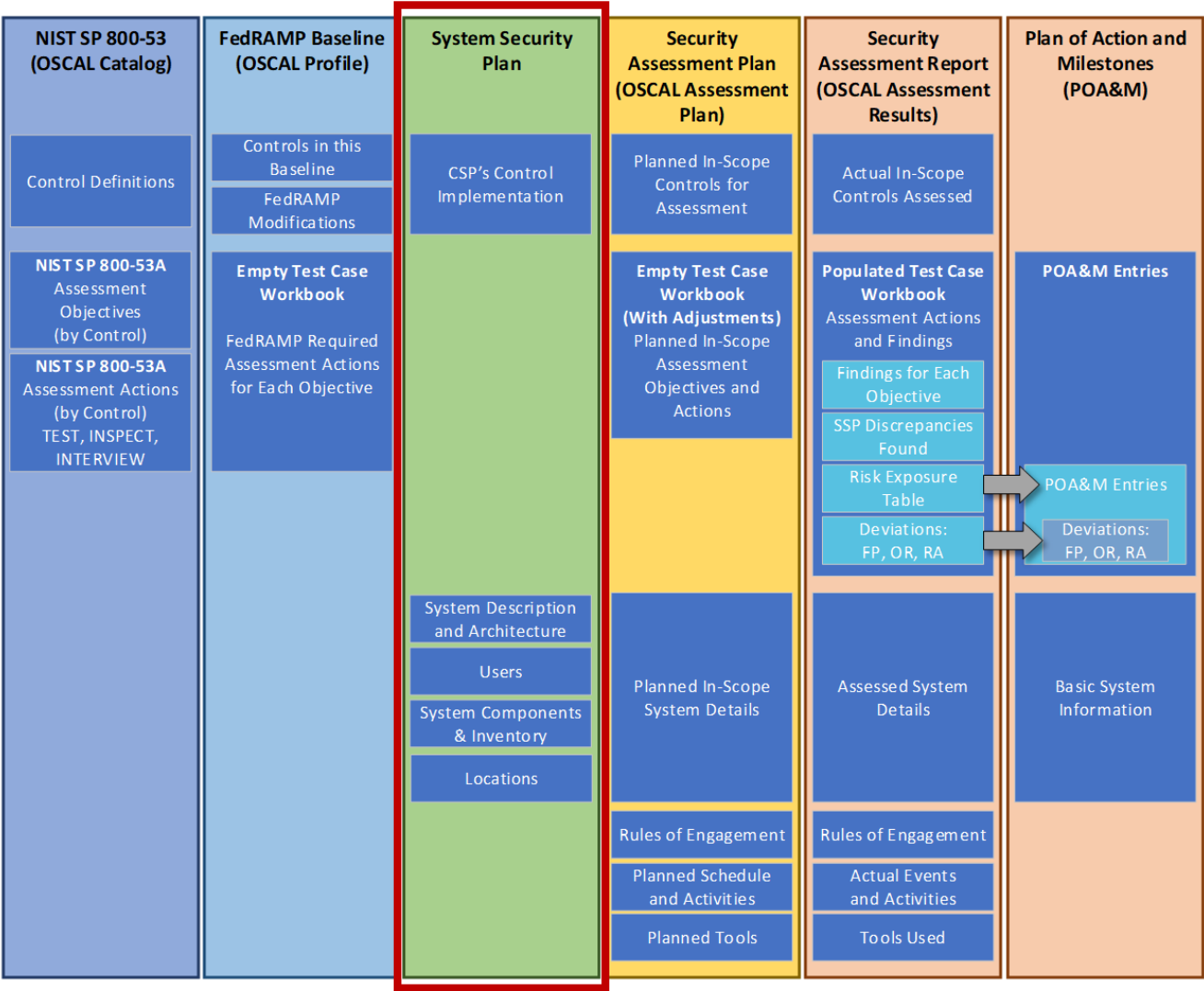
SSP File Concepts

Unlike the traditional MS Word-based SSP, SAP, and Security Assessment Report (SAR), the OSCAL-based versions of these files are designed to make information available through linkages, rather than duplicating information. In OSCAL, these linkages are established through import commands.



Each OSCAL file imports information from the one before it

For example, the NIST control definitions and FedRAMP baseline content that normally appears in Chapter 13 of the SSP are defined in the FedRAMP profile and simply referenced by the SSP.



Baseline Information is referenced instead of duplicated.

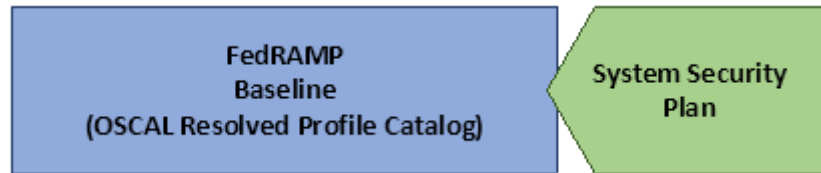
For this reason, an OSCAL-based SSP points to the appropriate OSCAL-based FedRAMP baseline as determined by the system's FIPS-199 impact level. Instead of duplicating control details, the OSCAL-based SSP simply points to the baseline content for information such as control definition statements, FedRAMP-added guidance, parameters, and FedRAMP-required parameter constraints.

Resolved Profile Catalogs

The resolved profile catalog for each FedRAMP baseline is a pre-processing of the profile and catalog to produce the resulting data. This reduces overhead for tools by eliminating the need to open and follow references from the profile to the catalog. It also includes only the catalog information relevant to the baseline, reducing the overhead of opening a larger catalog.

Where available, tool developers have the option of following the links from the profile to the catalog as described above, or using the resolved profile catalog.

Developers should be aware that at this time, catalogs and profiles remain relatively static. As OSCAL gains wider adoption, there is a risk that profiles and catalogs will become more dynamic, and a resolved profile catalog becomes more likely to be out of date. Early adopters may wish to start with the resolved profile catalog now, and plan to add functionality later for the separate profile and catalog handling later in their product roadmap.



The Resolved Profile Catalog for each FedRAMP Baseline reduces tool processing

For more information about resolved profile catalogs, see the [Guide to OSCAL-based FedRAMP Content Appendix C, Profile Resolution](#).

OSCAL-based FedRAMP SSP Template FedRAMP offers an OSCAL-based SSP shell file in both XML and JSON formats. This shell contains many of the FedRAMP required standards to help get you started. This document is intended to work in concert with that shell file. The OSCAL-based FedRAMP SSP Template is available in XML and JSON formats here:

- OSCAL-based FedRAMP SSP Template (JSON Format): <https://github.com/GSA/fedramp-automation/raw/master/dist/content/templates/ssp/json/FedRAMP-SSP-OSCAL-Template.json>
- OSCAL-based FedRAMP SSP Template (XML Format): <https://github.com/GSA/fedramp-automation/raw/master/dist/content/templates/ssp/xml/FedRAMP-SSP-OSCAL-Template.xml>

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

<https://pages.nist.gov/OSCAL/documentation/schema/implementation-layer/ssp>

Importing the FedRAMP Baseline

OSCAL is designed for traceability. Because of this, the SSP is designed to be linked to the FedRAMP baseline. Rather than duplicating content from the baseline, the SSP is intended to reference the baseline content itself.

Use the import-profile field to specify an existing OSCAL-based SSP. The href flag may include any valid uniform resource identifier (URI), including a relative path, absolute path, or URI fragment.

SSP Import Representation

```
<import-profile href="path/to/profile.xml" />
```

- OR -

```
<import-profile href="#[uuid-value]" />
```

XPath Queries

```
(SSP) URI to Baseline: /\*/import-profile/@href
```

If the value is a URI fragment, such as #96445439-6ce1-4e22-beae-aa72cfe173d0, the value to the right of the hashtag (#) is the UUID value of a resource in the SSP file's back-matter. Refer to the [*Guide to OSCAL-based FedRAMP Content, Section 2.6, Citations, Attachments and Embedded Content in OSCAL Files](#), for guidance on handling.

SSP Back Matter Representation

```
<back-matter>

  <resource id="96445439-6ce1-4e22-beae-aa72cfe173d0">

    <title>FedRAMP Moderate Baseline</title>

    <prop name="type" value="baseline"/>

    <!-- Specify the XML **or** JSON file location. Only one required. -->

    <rlink media-type="application/xml" href="./CSP\System\_SSP.xml" />

    <rlink media-type="application/json" href="./CSP\System\_SSP.json" />

  </resource>

</back-matter>
```

XPath Queries

(SSP) Referenced OSCAL-based FedRAMP Baseline

XML:

```
/*/back-matter/resource[@uuid='96445439-6ce1-4e22-beae-aa72cfe173d0'] /rlink[@media-type='application/xml']/@href
```

OR JSON:

```
/*/back-matter/resource[@uuid='96445439-6ce1-4e22-beae-aa72cfe173d0'] /rlink[@media-type='application/json']/@href
```