FedRAMP Security Controls

Control Response: Approaches

OSCAL offers a great deal of flexibility for controls responses. To balance consistency, interoperability and ease of adoption, the OSCAL Foundation recommends two approaches:

- **Flat Approach**: Aligns with FedRAMP's SSP Word template where control responses are at the statement level, and the narrative alone distinguishes between compoents within the response.
- **Normalized Approach**: Control responses are decomposed to align with relevant components.

With the **flat approach**, the entire statement-level response from a FedRAMP Word-based SSP is represented "as-is" in a single `by-component` assembly in OSCAL.

See Control Response: Flat Approach for more information.

> ## Retrofit Adoption Path: MVP
>
> If you have an existing FedRAMP authorization with an existing Word-based FedRAMP SSP, start with the flat approach and migrate over time to the normalized approach.

With the **normalized approach**, components are associated with control response statements. Responses are possible either for the whole statement or assocaited with a specific component relative to the statement response.

See Control Response: Normalized Approach for more information.

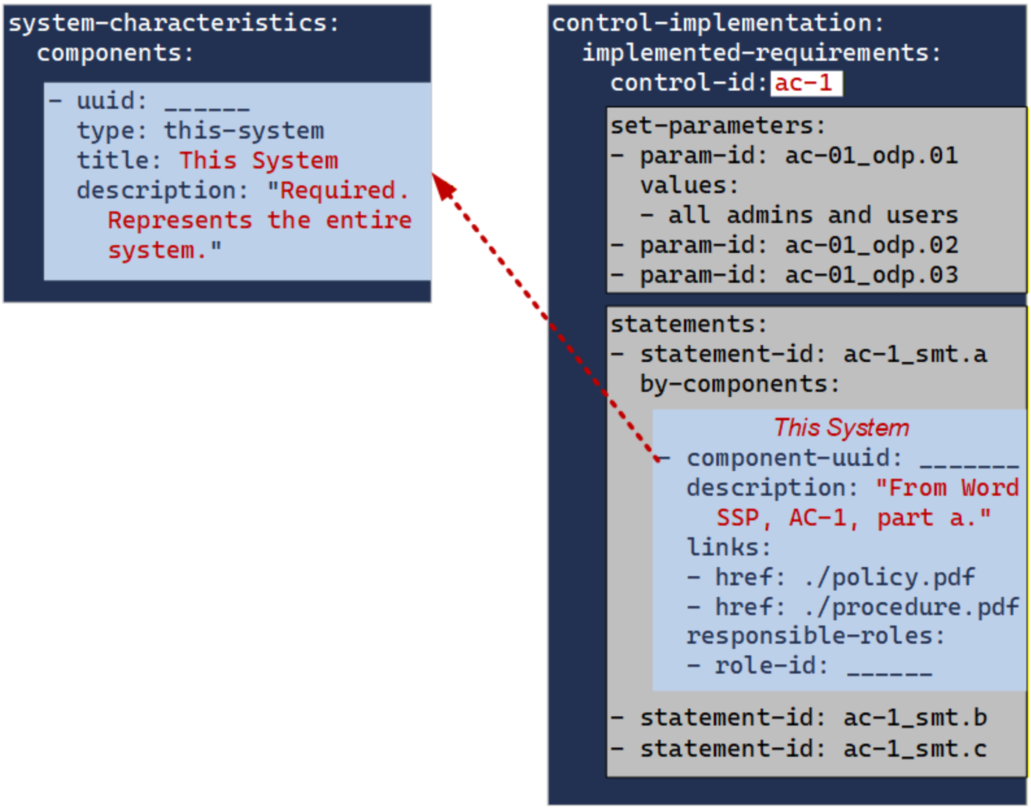
> ## New Adoption Path: Core
>
> If you are adopting OSCAL at the beginning of your FedRAMP journey, respond to control statements at the component level as much as practical. Define OSCAL components ahead of time, and be prepared to add components as needed for control response authoring.

Control Response: Flat Approach

The flat approach to control responses is only intended as a starting point for service providers converting from a legacy FedRAMP SSP Word template.

If you are not converting a legacy SSP, use the [Control Response: Normalized Approach](#).

With the flat approach, the entire statement-level response from a FedRAMP Word-based SSP is represented "as-is" in a single `by-component` entry in OSCAL.



Retrofit Adoption Path: MVP

With OSCAL SSPs, all control responses must be assocaited with a component. To ensure this is always possible, OSCAL SSPs also require the existence of a `this system` component, which represents the entire system.

When converting from a legacy Word-based SSP, the simpelest form of OSCAL adoption is to move the text from each control statement response into the "this system" component response.

Transition to Normalized

Over time, components can be added to the `components` array in `system-characteristics`. Some components will be added in order to represent SSP tables, such as leveraged authorizations, external services and cryptographic modules. Others may be added to support [inventory normalization](). Add any additional components you need to support or control responses.

At any time, additional `by-components` entries can be added to a `statements` entry, and linked to a component. This may occur one component at a time.

Example Transition

The legacy Word-Based SSP, response to AC-1, Statement a is:

> *The Trust an Compliance Team developed, maintains and disseminates the XYZ Corp Access Control Policy, v2.3 dated January 5th 2024 to all management, administrators and users of the PDQ Cloud System.*
>
> *Chapters 1 and 2 define purpose and scope, while chapter 3 defines roles. Chapters 4 - 8 define responsibilities and coordination, and chapter 9 confirms maangement commitment and potential penalties.*
>
> *The PDQ Information System Security Officer developed, maintains and disseminates the PDQ Access Control Procedure, v 1.1 dated March 1, 2026, which defines access control operations for the system. The ISSO ensures all PDQ Cloud System managers and administrators receive a copy of this docuemnt*

MVP OSCAL Representation

The entire statement above is represented as follows:

- `metadata/roles` entries for the ISSO and Trust and Compliance Team.
- a `this-system` entry in the `components` array
- an `implemented-requirements` entry for AC-1 (`ac-1`)
  - a `statements` entry for AC-1, part a (`ac-1_smt.a`)
    - a `by-components` entry with the `component-uuid` value of the `this-system` entry in the `components` array
      - a `description` field with the statement from the Word-based SSP.

```
system-security-plan:
  metadata:
    roles:
    - role-id: information-system-security-officer
      title: ISSO
    - role-id: trust-and-compliance
      title: Corporate Trust and Compliance Team
  system-implementation:
    components:
    - uuid: 11111111-2222-4000-8000-009000000000
      type: this-system
      title: This System
      description: 'This component represents the entire system or authorization boundary.'

  control-implementation:
    description: 'OSCAL-required field.'
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-1

      statements:
      - statement-id: ac-1_smt.a
        uuid: 11111111-2222-4000-8000-012000010100
        by-components:
        - component-uuid: 11111111-2222-4000-8000-009000000000
          uuid: 11111111-2222-4000-8000-012000010101

          description: 'The Trust an Compliance Team developed, maintains and disseminates the
XYZ Corp Access Control Policy, v2.3 dated January 5th 2024 to all management, administrators
and users of the PDQ Cloud System.

Chapters 1 and 2 define purpose and scope, while chapter 3 defines roles. Chapters 4 - 8
define responsibilities and coordination, and chapter 9 confirms maangement commitment and
potential penalties.

The PDQ Information System Security Officer developed, maintains and disseminates the PDQ
Access Control Procedure, v 1.1 dated March 1, 2026, which defines access control operations
for the system. The ISSO ensures all PDQ Cloud System managers and administrators receive a
```

```
copy of this docuemnt.'

          implementation-status:
            state: implemented
          responsible-roles:
          - role-id: information-system-security-officer
          - role-id: trust-and-compliance
```

Transition

In moving to the *normalized* approach, OSCAL components must eventually be defined for required documents. This will result in additional entries to the `components` array as follows:

- Additional entries to the `components` array
  - a `type` set to `policy` or `process-procedure`
  - a `title` with the title of the policy or procedure
  - a `responsible-roles` array with the appropraite role-id cited.

```
system-security-plan:
  system-implementation:
    components:
    - uuid: 11111111-2222-4000-8000-009000000001
      type: policy
      title: XYZ Access Control Policy
      description: 'This is the corporate AC Policy.'
      responsible-roles:
      - role-id: trust-and-compliance

    - uuid: 11111111-2222-4000-8000-009000000003
      type: policy
      title: PDQ Access Control Procedure
      description: 'This is the system-specific AC Procedure.'
      responsible-roles:
      - role-id: information-system-security-officer
```

Once defined, additional by-component entries may be added to the AC-1, part a atatement; however they do not need to be added all at once. For example, the policy may be addressed in one pass and the procedures deferred.

- add one additional `by-components` entry for the policy
- move only the policy portion of the control response
- drop the `trust-and-compliance` role
  - It is not necessary to move the `trust-and-compliance` role as it is defined for the component above.

```
system-security-plan:
  control-implementation:
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-1
      statements:
      - statement-id: ac-1_smt.a
        uuid: 11111111-2222-4000-8000-012000010100
        by-components:

        - component-uuid: 11111111-2222-4000-8000-009000000000
          uuid: 11111111-2222-4000-8000-012000010101

          description: 'The PDQ Information System Security Officer developed, maintains and disseminates the PDQ Access Control Procedure, v 1.1 dated March 1, 2026, which defines access control operations for the system. The ISSO ensures all PDQ Cloud System managers and administrators receive a copy of this docuemnt.'

          implementation-status:
            state: implemented
          responsible-roles:
          - role-id: information-system-security-officer

        - component-uuid: 11111111-2222-4000-8000-009000000001
          uuid: 11111111-2222-4000-8000-012000010102

          description: 'The Trust an Compliance Team developed, maintans and disseminates the XYZ Corp Access Control Policy, v2.3 dated January 5th 2024 to all management, administrators and users of the PDQ Cloud System.

Chapters 1 and 2 define purpose and scope, while chapter 3 defines roles. Chapters 4 - 8 define responsibilities and coordination, and chapter 9 confirms maangement commitment and potential penalties.'
```

```yaml
          implementation-status:
            state: implemented
```

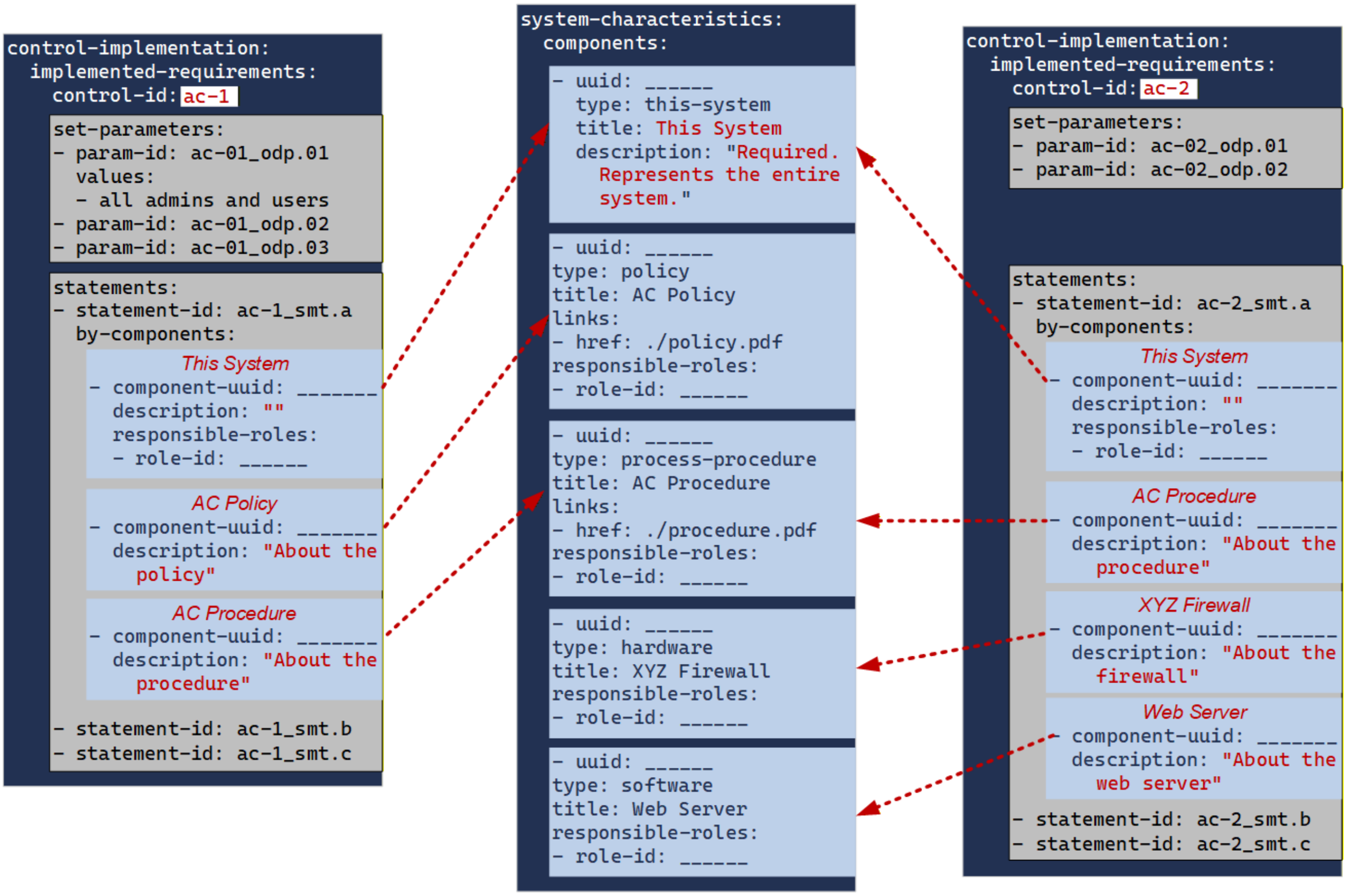
When all components have been added, the original `by-components` entry for `this-system` may still be used for providing information (control responses, status differences or additional roles) that do not fit specific component responses.

Control Response: Normalized Approach

The normalized approach is prefered. Organizations starting new with no legacy SSP content should use this.

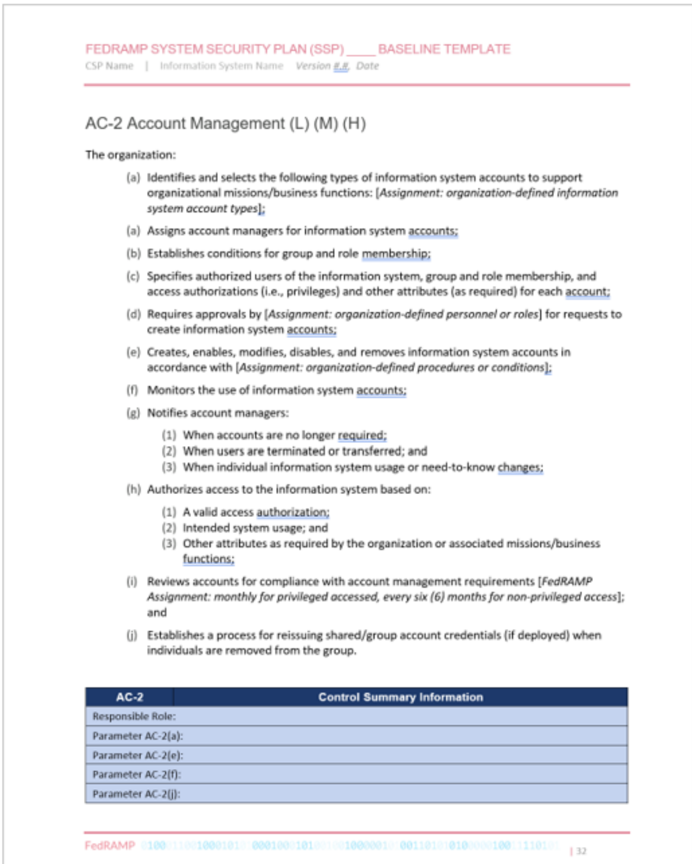
For organizations converting from a legacy FedRAMP SSP Word template, consider starting with the [Control Response: Flat Approach] and migrating to the normalized approach over time.

With the normalized approach, system elements are first defined as OSCAL `components`. Relvant components are then associated with control statements via `statements`/`by-components` entries. Control responses are then provided in the apprpark `by-component` entry.




```
system-security-plan:
```

Responding to Control Baselines



FEDRAMP SYSTEM SECURITY PLAN (SSP) ____ BASELINE TEMPLATE
CSP Name | Information System Name Version #.#, Date

AC-2 Account Management (L) (M) (H)

The organization:

(a) Identifies and selects the following types of information system accounts to support organizational missions/business functions: [*Assignment: organization-defined information system account types*];

(a) Assigns account managers for information system accounts;

(b) Establishes conditions for group and role membership;

(c) Specifies authorized users of the information system, group and role membership, and access authorizations (i.e., privileges) and other attributes (as required) for each account;

(d) Requires approvals by [*Assignment: organization-defined personnel or roles*] for requests to create information system accounts;

(e) Creates, enables, modifies, disables, and removes information system accounts in accordance with [*Assignment: organization-defined procedures or conditions*];

(f) Monitors the use of information system accounts;

(g) Notifies account managers:

(1) When accounts are no longer required;
(2) When users are terminated or transferred; and
(3) When individual information system usage or need-to-know changes;

(h) Authorizes access to the information system based on:

(1) A valid access authorization;
(2) Intended system usage; and
(3) Other attributes as required by the organization or associated missions/business functions;

(i) Reviews accounts for compliance with account management requirements [*FedRAMP Assignment: monthly for privileged accessed, every six (6) months for non-privileged access*]; and

(j) Establishes a process for reissuing shared/group account credentials (if deployed) when individuals are removed from the group.

AC-2	Control Summary Information
Responsible Role:	
Parameter AC-2(a):	
Parameter AC-2(e):	
Parameter AC-2(f):	
Parameter AC-2(j):	

FedRAMP | 32

OSCAL references controls in baselines and catalogs. The statements are not duplicated into an OSCAL SSP the way they are with a Word SSP.

Conrol baseline requirements are imported by an OSCAL SSP and referenced as needed.

Importing a Baseline

Import the appropriate FedRAMP Baseline, either as an OSCAL *profile* or as an OSCAL *reserved profile catalog*.

```yaml
system-security-plan:
  import-profile:
    href: https://raw.githubusercontent.com/OSCAL-Foundation/fedramp-resources/refs/heads/main/baselines/rev5/yaml/FedRAMP_rev5_HIGH-baseline-resolved-profile_catalog.yaml
```

> The OSCAL Foundation makes the FedRAMP baselines available as OSCAL _profiles_ and _resolved profile catalogs_ [on GitHub](https://github.com/OSCAL-Foundation/fedramp-resources/tree/main/baselines/rev5).
>
> See Baselines for more information about those files.

Referencing Controls

With the approprate baseline imported above, OSCAL SSP control responses simply cite the control `id` from the baseline.

For each control in the imported baseline there MUST be exactly one `implemented-requirements` entry that includes:

- a `uuid`
- a `control-id` with a value that matches a control in the imported baseline
- a `set-parameters` array, only if the control has one or more parameters that don't already have their `value` established in the baseline. See *Parameter Assignments* for more information.
- a `statements` array contains the control responses. See *Control Implementation Statements* for more information.

```
system-security-plan:
  control-implementation:
    description: 'This description field is required by OSCAL, but ignored by FedRAMP.'
    implemented-requirements:

    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-1
      set-parameters:
        [content cut]
      statements:
        [content cut]

    - uuid: 11111111-2222-4000-8000-012000010001
      control-id: ac-2
      [content cut]

    - uuid: 11111111-2222-4000-8000-012000010002
      control-id: ac-2.1
```

```
[content cut]
```

Responsible Roles

Every control should have one or more responsible roles identified.

◢ AU-5 Response to Audit Logging Process Failures (L)(M)(H)

a. Alert [Assignment: organization-defined personnel or roles] within [Assignment: organization-defined time period] in the event of an audit logging process failure; and

b. Take the following additional actions: [FedRAMP Assignment: overwrite oldest record].

AU-5 Control Summary Information
Responsible Role:
Parameter AU-5(a)-1:
Parameter AU-5(a)-2:
Parameter AU-5(b):
Implementation Status (check all that apply): ☐ Implemented ☐ Partially Implemented ☐ Planned ☐ Alternative implementation ☐ Not Applicable
Control Origination (check all that apply): ☐ Service Provider Corporate ☐ Service Provider System Specific ☐ Service Provider Hybrid (Corporate and System Specific) ☐ Configured by Customer (Customer System Specific) ☐ Provided by Customer (Customer System Specific) ☐ Shared (Service Provider and Customer Responsibility) ☐ Inherited from pre-existing FedRAMP Authorization for [Click here to enter text], Date of Authorization

In OSCAL, there are three possible sources for responsible roles:

- **By Control**: (Retrofit MVP only) assign responsible roles to the `implemented-requirement` for the entire control
- **By Component (Implied)**: infer responsible roles from the components cited in the `by-component` array
- **By Component (Explicit)**: assign responsible roles to the `statement`/`by-component` array

Retrofit Adoption Path: MVP

When initially converting a Word-based FedRAMP SSP to OSCAL, assign all roles *by control* to the `implemented-requirements`/`responsible-roles` array. This aligns with the FedRAMP Word-based SSP template.

As the SSP is migrated to a normalized approach using components, the assignment of roles is moved from the entire control to statement-level, component responses.

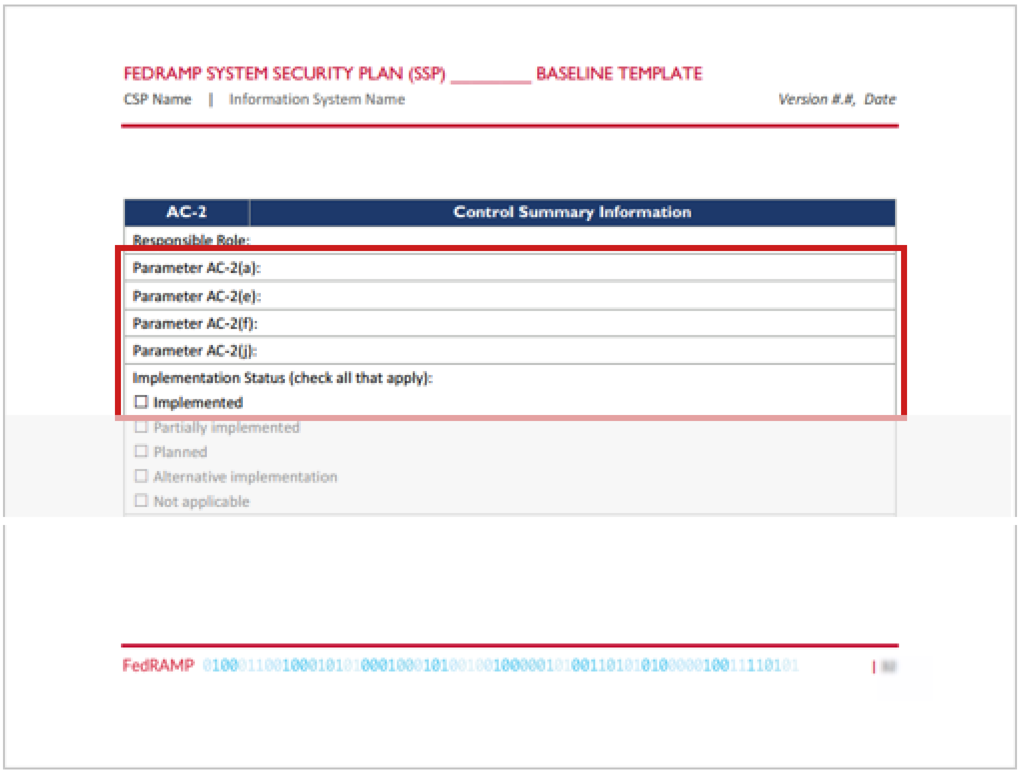
With fully normalized OSCAL content, responsible roles are inferred via the components associated with a control via `statements`/`by-components`. Each assocaited component SHOULD have `owner` and `administrator` responsible roles and linked to specific parties (teams or individuals).

If additional roles need to be cited, they are explicilty assigned to `by-components`/`responsible-roles`. If an explicitly needed role does not associate cleanly to a specific component, it is assigned to the `by-components`/`responsible-roles` entry for *this system* (component `type`=`this-system`).

WORKING HERE

Representation

Parameter Assignments



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

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

AC-2	Control Summary Information
Responsible Role:	
Parameter AC-2(a):	
Parameter AC-2(e):	
Parameter AC-2(f):	
Parameter AC-2(j):	
Implementation Status (check all that apply): ☐ Implemented ☐ Partially implemented ☐ Planned ☐ Alternative implementation ☐ Not applicable	

FedRAMP

Representation

If a FedRAMP control has one or more parameters, add a `set-parameters` array Within an `implemented-requirements` entry. There must be one `set-parameters` entry for each parameter in the control as follows:

- a `param-id` set to the parameter value from the OSCAL-based FedRAMP baselines
- a `values` array with:
  - one string entry per response
  - If the response is list, such as a list of user types to receive a procedure, add one entry per list item.

> Only set parameters at the `implemented-requirements` level. While OSCAL also supports the ability to set parameters within `by-components` entries, this does not align with FedRAMP's handling of parameters and should not be used.

```
system-security-plan:
  control-implementation:
    implemented-requirements:
```

```yaml
- uuid: 11111111-2222-4000-8000-012000010000
  control-id: ac-1
  set-parameters:

  - param-id: ac-01_odp.01
    values:
    - all managers, administrators and users of the system

  - param-id: ac-01_odp.02
    values:
    - all managers and administrators of the system

  - param-id: ac-01_odp.03
    values:
    - System-level

  - param-id: ac-01_odp.04
    values:
    - System Architect

  - param-id: ac-01_odp.05
    values:
    - at least every 3 years

  - param-id: ac-01_odp.06
    values:
    - change in organizational legal status or ownership

  - param-id: ac-01_odp.07
    values:
    - at least annually

  - param-id: ac-01_odp.08
    values:
    - change in policy or a security incident involving a failure of access control
      mechanisms
```

Selection Parameters and Nested Parameters

Some *select* parameters contain one or more *assignment* parameters. In this instance, simply provide the final selection value within the `set-parameters` entry for the *select* and omit any `set-parameters` entries related to the *assignment*.

Example

AC-7_ part *(b)* has three *assignment* parameters nested within a single *selection* parameter. Line breaks and bullets have been added below to better illustrate the nesting.

> *Automatically*
>
> - *[**Selection** (one or more):*
>   - *lock the account or node for an [**Assignment**: organization-defined time period];*
>   - *lock the account or node until released by an administrator;*
>   - *delay next logon prompt per [**Assignment**: organization-defined delay algorithm];*
>   - *notify system administrator;*
>   - *take other [**Assignment**: organization-defined action]]*
>
> *when the maximum number of unsuccessful attempts is exceeded.*

Although the OSCAL controls will have four parameters, only the final value for the *selection* parameter is assigned in the SSP. The other parameters are ignored.

If more than one choice is is applicable, add each as a separate entry in the `values` array. For example if the final choices are:

- lock the account or node for an [**Assignment**: 30 minutes];
- lock the account or node until released by an administrator;

The `set-parameters` array would be:

```
system-security-plan:
```

```
  control-implementation:
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-7
      set-parameters:

      - param-id: ac-07_odp.03
        values:
        - lock the account or node for 30 minutes;
        - lock the account or node until released by an administrator;
```

Parameters `ac-07_odp.01` and `ac-07_odp.02` belong to part (a). They would normally be included and are only omitted for the example.

Parameters `ac-07_odp.04`, `ac-07_odp.05` and `ac-07_odp.06` are part of `ac-07_odp.03` and are omitted.

Implementaiton Status

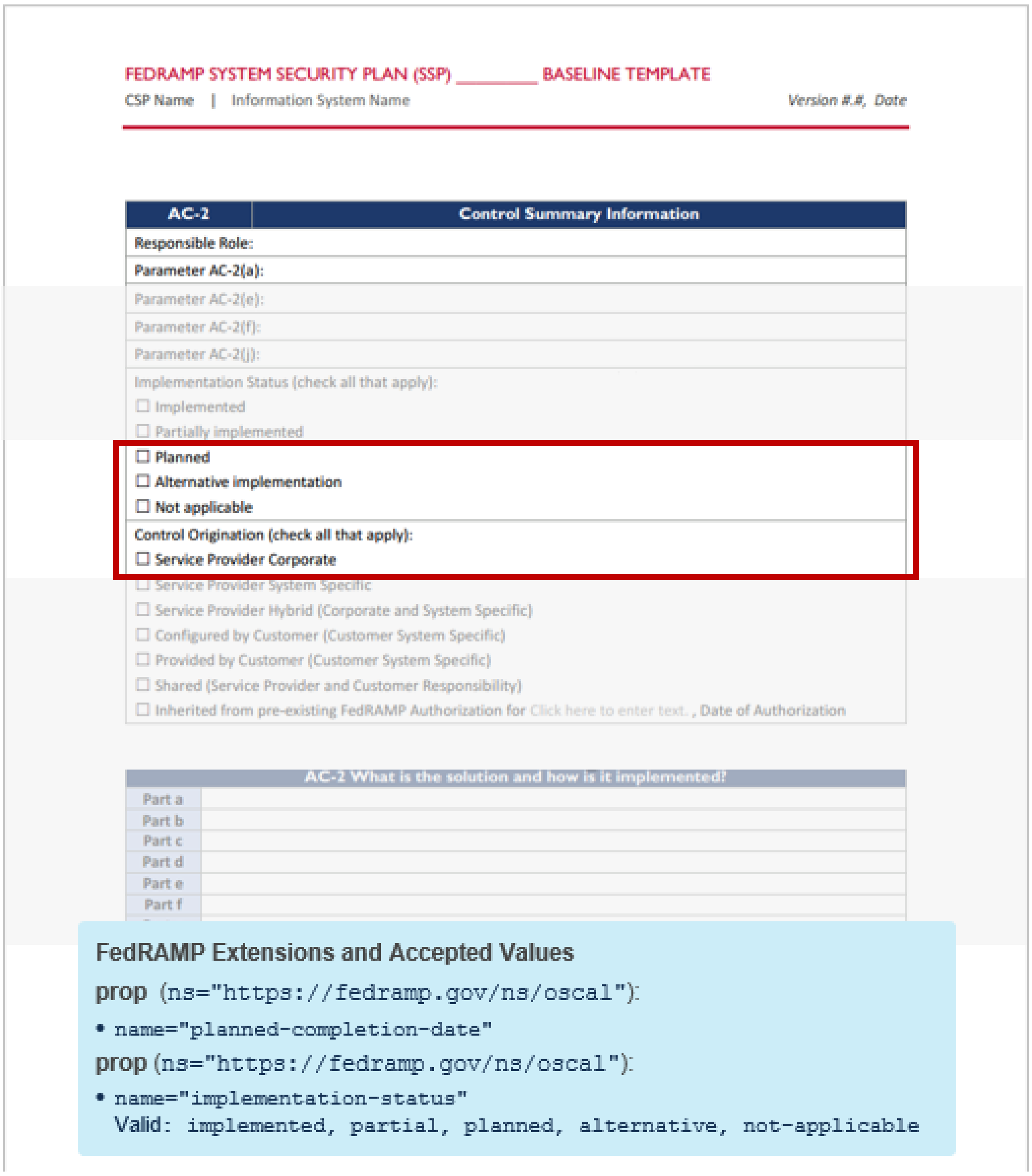
FedRAMP only accepts only one of five values for `implementation-status`: implemented, partial, planned, alternative, and not-applicable. A control may be marked "partial" and "planned" (using two separate implementation-status fields). All other choices are mutually exclusive.

If the implementation-status is partial, the gap must be explained in the remarks field.

If the implementation-status is planned, a brief description of the plan to address the gap, including major milestones must be explained in the `remarks` field. There must also be a prop (name="planned-completion-date" ns="http://fedramp.gov/ns/oscal") field containing the intended completion date. With XML, `prop` fields must appear before other sibling fields (such as `set-parmeter`, `responsible-role`, etc.), even though that sequence is counter-intuitive in this situation.

If the implementation-status is alternative, the alternative implementation must be summarized in the `remarks` field.

If the implementation-status is not-applicable, the N/A justification must be provided in the `remarks` field.



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE

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

AC-2	Control Summary Information
Responsible Role:	
Parameter AC-2(a):	
Parameter AC-2(e):	
Parameter AC-2(f):	
Parameter AC-2(j):	
Implementation Status (check all that apply): ☐ Implemented ☐ Partially implemented ☐ Planned ☐ Alternative implementation ☐ Not applicable	
Control Origination (check all that apply): ☐ Service Provider Corporate ☐ Service Provider System Specific ☐ Service Provider Hybrid (Corporate and System Specific) ☐ Configured by Customer (Customer System Specific) ☐ Provided by Customer (Customer System Specific) ☐ Shared (Service Provider and Customer Responsibility) ☐ Inherited from pre-existing FedRAMP Authorization for Click here to enter text. , Date of Authorization	

AC-2 What is the solution and how is it implemented?	
Part a	
Part b	
Part c	
Part d	
Part e	
Part f	

FedRAMP Extensions and Accepted Values

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

- `name="planned-completion-date"`

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

- `name="implementation-status"`
  Valid: `implemented, partial, planned, alternative, not-applicable`

Representation

```
<!-- system-implementation -->
<control-implementation>
    <implemented-requirement uuid="uuid-value" control-id="ac-1">
        <prop name="planned-completion-date"
            ns="http://fedramp.gov/ns/oscal" value="2021-01-01Z"/>
        <prop name="implementation-status"
            ns="http://fedramp.gov/ns/oscal" value="implemented" />
        <prop name="implementation-status"
            ns="http://fedramp.gov/ns/oscal" value="partial" />
        <prop name="implementation-status"
            ns="http://fedramp.gov/ns/oscal" value="planned" />
        <prop name="implementation-status"
```

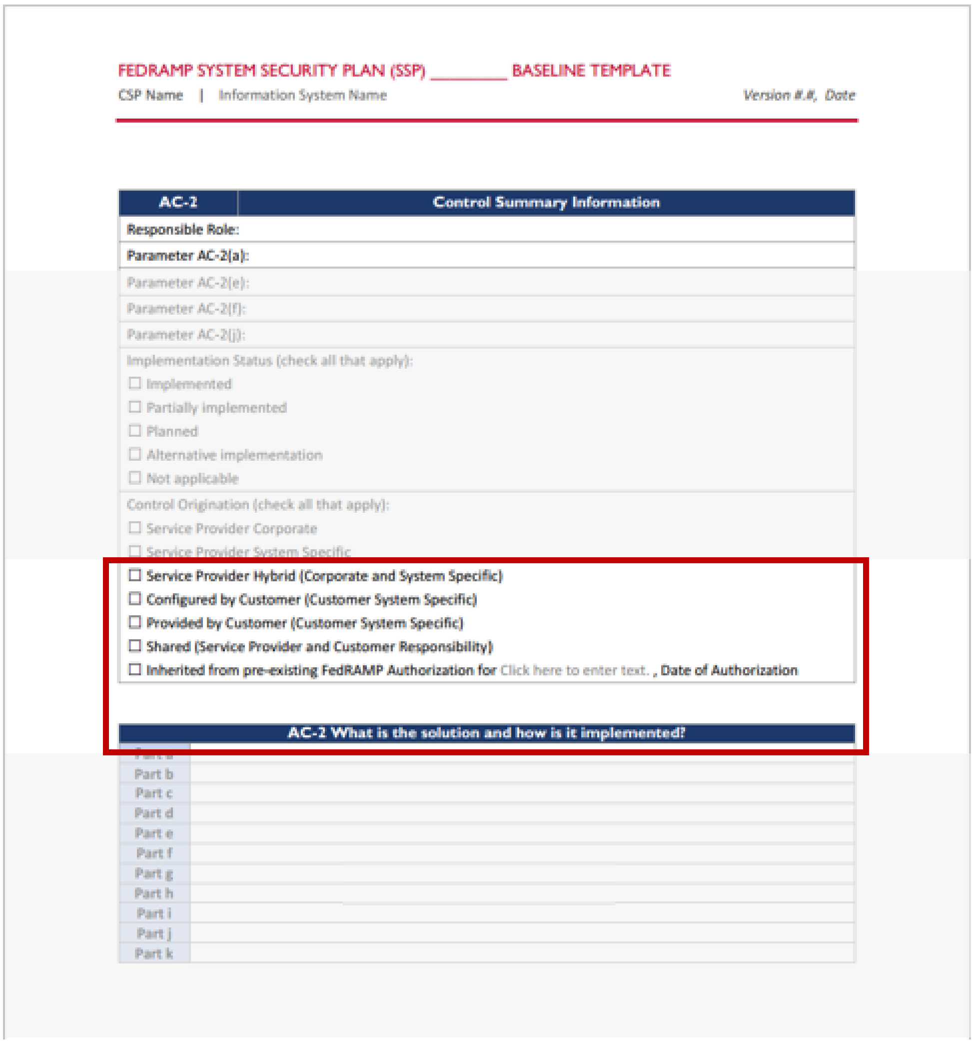
```
            ns="http://fedramp.gov/ns/oscal" value="not-applicable"/>
        <!-- responsible-role, statement, by-component -->
    </implemented-requirement>
</control-implementation>
<!-- back-matter -->
```

The FedRAMP `implementation-status` property at the control's `implemented-requirement` level is a summary of all statement and/or component level core OSCAL `implementation-status` designations. It must be set appropriately based on the least value of child statement or component level `implementation-status` designations. When a statement and/or component level `implementation-status` designation is not specified, the FedRAMP `implementation-status` value is assumed. Individual statements and/or components may override `implementation-status` locally.

Control Origination

FedRAMP accepts only one of five values for `control-origination`: sp-corporate, sp-system, customer-configured, customer-provided, and inherited. Hybrid choices are expressed by identifying more than one `control-origination`, each in a separate prop field.
For controls with a control-id ending in "-1", FedRAMP only accepts sp-corporate and sp-system.

If the control origination is inherited, there must also be a FedRAMP extension (prop name="leveraged-authorization-uuid" ns="http://fedramp.gov/ns/oscal") field containing the UUID of the leveraged authorization as it appears in the /*/system-implementation/leveraged-authorization assembly.



FEDRAMP SYSTEM SECURITY PLAN (SSP) ________ BASELINE TEMPLATE
CSP Name | Information System Name — Version #.#, Date

AC-2	Control Summary Information
Responsible Role:	
Parameter AC-2(a):	
Parameter AC-2(e):	
Parameter AC-2(f):	
Parameter AC-2(j):	
Implementation Status (check all that apply): ☐ Implemented ☐ Partially implemented ☐ Planned ☐ Alternative implementation ☐ Not applicable	
Control Origination (check all that apply): ☐ Service Provider Corporate ☐ Service Provider System Specific ☐ Service Provider Hybrid (Corporate and System Specific) ☐ Configured by Customer (Customer System Specific) ☐ Provided by Customer (Customer System Specific) ☐ Shared (Service Provider and Customer Responsibility) ☐ Inherited from pre-existing FedRAMP Authorization for Click here to enter text. , Date of Authorization	

AC-2 What is the solution and how is it implemented?	
Part a	
Part b	
Part c	
Part d	
Part e	
Part f	
Part g	
Part h	
Part i	
Part j	
Part k	

Representation

```xml
<system-implementation>
    <!-- status -->
    <leveraged-authorization uuid="uuid-of-leveraged-authorization">
        <!-- details cut - see Leveraged Authorizations Section -->
    </leveraged-authorization>
</system-implmentation>

<control-implementation>
    <implemented-requirement uuid="uuid-value" control-id="ac-2">
        <prop name="leveraged-authorization-uuid"
            value="uuid-of-leveraged-authorization"/>
        <prop ns="http://fedramp.gov/ns/oscal" name="control-origination"
            value="sp-corporate" />
        <prop ns="http://fedramp.gov/ns/oscal" name="control-origination"
            value="sp-system" />
        <prop ns="http://fedramp.gov/ns/oscal" name="control-origination"
            value="customer-configured" />
        <prop ns="http://fedramp.gov/ns/oscal" name="control-origination"
            value="inherited" />
        <!-- responsible-role -->
    </implemented-requirement>
</control-implementation>
<!-- back-matter -->
```

Responding By Component

AC-1 What is the solution and how is it implemented?
Part a:
Part b:
Part c:

OSCAL SSPs represent control responses in `control-implementation` / `implemented-requirements` / `statements`.

See Control Implementation Statements to understand how to associate control responses with specific baseline controls and control statements.

Within `statements`, all responses must be assocaited with one or more components via the `by-components` array.

OSCAL enables you to be as granular as you wish. Individual components may be added for operating systems, container images, firewalls, policies, procedures and plans. There is always a "this-system" component representing the entire system / authorization-boundary.

The "This System" Component

There must always be a **"This System"** component defined in the SSP. For control responses, this is used in several ways:

- **Holistic Overview**: The SSP author may wish to provide a more holistic overview of how several components work together, even if details are provided individually in other `by-component` assemblies.
- **Catch-all**: Any control response that does not cleanly align with another system component may be described in the **"This System"** component.
- **Legacy SSP Conversion**: When converting a legacy SSP to OSCAL, the legacy control response statements may initially be associated with

> the **"This System"** component until the SSP author is able to provide responses for individual components.

responses occur within `by-components` / `description`. In a legacy Word-based SSP, it was often necessary to provide narriative for each relevant component in a control response. The entire narriative for all components was captured in a single table cell as separate paragraphs.

With OSCAL, you have the option of keeping a single narriative block, or breaking out a control response by its discrete components.

Retrofit Adoption Path MVP

When converting a Word-based FedRAMP SSP to OSCAL, move all control responses to the `this-system` component.

Every OSCAL SSP must have a `this-system` component defined. It is the only required component.

```
system-security-plan:
  system-implementation:
    components:
    - uuid: 11111111-2222-4000-8000-009000000000
      type: this-system
      title: This System
      description: 'Represents the entire authorization boundary'
      status:
        state: operational
```

Every `statements` / `by-components` array has exactly one entry that references the `this-system` component and includes the content from the Word-based SSP.

Each `statements` array entry includes:

- a required `uuid` field
- a required `by-components` array. Each array entry includes:
  - a required `component-uuid` field that cites the `this-sytem` component from above.
  - a required `uuid` field

    - a required `description` field that contains the content from the Word-based SSP control response.
    - a required `implementation-status` element with:
        - a required `state` field with a value of of `implemented`.


```yaml
system-security-plan:
  control-implementation:
    description: n/a.
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-1
      statements:
      - statement-id: ac-1_smt.a
        uuid: 11111111-2222-4000-8000-012000010100
        by-components:
        - component-uuid: 11111111-2222-4000-8000-009000000000
          uuid: 11111111-2222-4000-8000-012000010101
          description: Word-based SSP AC-1, statement a response.
          implementation-status:
            state: implemented
      - statement-id: ac-1_smt.b
        uuid: 11111111-2222-4000-8000-012000010200
        by-components:
        - component-uuid: 11111111-2222-4000-8000-009000000000
          uuid: 11111111-2222-4000-8000-012000010201
          description: Word-based SSP AC-1, statement b response.
      - statement-id: ac-1_smt.c
        uuid: 11111111-2222-4000-8000-012000010300
        by-components:
        - component-uuid: 11111111-2222-4000-8000-009000000000
          uuid: 11111111-2222-4000-8000-012000010301
          description: Word-based SSP AC-1, statement c response.
          implementation-status:
            state: implemented
```


> See the Example below.

Native Adoption Path

When creating an SSP from scratch, ensure appropriate components are defined before authoring a control response. The `this-system` component must always be present. Other components are present baed on their use within the sytem. See [Components](#) for more information.

```yaml
system-security-plan:
  system-implementation:
    components:
    - uuid: 11111111-2222-4000-8000-009000000000
      type: this-system
      title: This System
      description: 'Represents the entire authorization boundary'
      status:
        state: operational

    - uuid: 11111111-2222-4000-8000-009000600001
      type: policy
      title: Access Control and Identity Management Policy
      description: 'A corporate policy used for the system.'
      status:
        state: operational
```

Every `statements` / `by-components` array has one or more entries that reference components describes how that component is satisfying that control requirement statement.

Each `statements` array entry includes:

- a required `uuid` field
- a required `by-components` array. Each array entry includes:
  - a required `component-uuid` field that cites the appropriate component from above.
  - a required `uuid` field
  - a required `description` field that contains the content from the Word-based SSP control response.
  - a required `implementation-status` element with:
    - a required `state` field with a value of of `implemented`.

```yaml
system-security-plan:
  control-implementation:
    description: n/a.
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-1
      statements:
      - statement-id: ac-1_smt.a
        uuid: 11111111-2222-4000-8000-012000010100
        by-components:

        - component-uuid: 11111111-2222-4000-8000-009000600001
          uuid: 11111111-2222-4000-8000-012000010102
          description: Describe how this policy satisfies part a.
          implementation-status:
            state: implemented

        - component-uuid: 11111111-2222-4000-8000-009000000000
          uuid: 11111111-2222-4000-8000-012000010101
          description: "Provide general context about satisfying part a that doesn't fit a
defined component."
          implementation-status:
            state: implemented
```

Example

IA-2 Identificaiton and Authentication (Organizational Users) is satisfied by a combination of:

- the IA Policy
- an IA Procedure
- a container running KeyCloak
- an enterprise directory capability

This was originally described in the the IA-2 narrative as:

> All components requiring authentication are configured to redirect users to KeyCloak. When a user supplies their ID and KeyCloak recognizes it as belonging to this organization, it redirects the user's authentication attempt to the enterprise directory capability for authentication. The enterprise directory reports the user's authentication success or failure back to KeyCloak. If authentication is successful, KeyCloak generates an access token and passes it back to the compnent requesting authentication.
>
> The IA Policy requires use of the enterprise directory for authentication of organizational users. The system-level IA Procedure provides instructions for admins to configure their compoents to use KeyCloak for authentication.

Within the OSCAL SSP, this entire statement can initially be assocaited with the "this-system" component in the `by-component` response to AC-2.

- `by-component` (`this-system`)

> All components requiring authentication are configured to redirect users to KeyCloak. When a user supplies their ID and KeyCloak recognizes it as belonging to this organization, it redirects the user's authentication attempt to the enterprise directory capability for authentication. The enterprise directory reports the user's authentication success or failure back to KeyCloak. If authentication is successful, KeyCloak generates an access token and passes it back to the compnent requesting authentication.
>
> The IA Policy requires use of the enterprise directory for authentication of organizational users. The system-level IA Procedure provides instructions for admins to configure their compoents to use KeyCloak for authentication.

Moving Toward Normalization

At a later date, the SSP author can define components for the IA Policy and system-level IA Procedure and associate them with AC-2. The content shifts to be represented like this:

- `by-component` (`this-system`)

  > All components requiring authentication are configured to redirect users to KeyCloak. When a user supplies their ID and KeyCloak recognizes it as belonging to this organization, it redirects the user's authentication attempt to the enterprise directory capability for authentication. The enterprise directory reports the user's authentication success or failure back to KeyCloak. If authentication is successful, KeyCloak generates an access token and passes it back to the compnent requesting authentication.

- `by-component` (`policy`)

  > The IA Policy requires use of the enterprise directory for authentication of organizational users.

- `by-component` (`process-procedure`)

  > The system-level IA Procedure provides instructions for admins to configure their compoents to use KeyCloak for authentication.

Fully Normalized

Eventually, components are added for KeyCloak and the enterprise directory; however, some of this narrative describes how the two work together. The `this-system` component can still be used for any narrative that doesn't fit cleanly in another component.

- `by-component` (`this-system`)

> All components requiring authentication are configured to redirect users to KeyCloak.

- `by-component` (`software` / KeyCloak)

  > When a user supplies their ID and KeyCloak recognizes it as belonging to this organization, it redirects the user's authentication attempt to the enterprise directory capability for authentication.
  >
  > If authentication is successful, KeyCloak generates an access token and passes it back to the compnent requesting authentication.

- `by-component` (`service` / enterprise directory)

  > The enterprise directory reports the user's authentication success or failure back to KeyCloak.

- `by-component` (`policy`)

  > The IA Policy requires use of the enterprise directory for authentication of organizational users.

- `by-component` (`process-procedure`)

  > The system-level IA Procedure provides instructions for admins to configure their compoents to use KeyCloak for authentication.

This is now fully normalized.

Control Implementation Statements

Typically, the controls in the FedRAMP baselines have lettered parts (a., b., etc.). A few only have a top-level statement with no parts. Current FedRAMP templates expect responses at the lettered part level when present and at the top-level otherwise.

OSCAL SSPs cite controls and control requirement statements in responses.

> Within the OSCAL FedRAMP baselines, each control statement is assigned an identifier. Any lettered parts are also assigned identifiers.
>
> Citing statement identifiers correctly is critical to automated processing.
>
> See Citing Control Statements for important information.

Typical

AC-1 What is the solution and how is it implemented?
Part a:
Part b:
Part c:

Most FedRAMP controls have two or more lettered parts. FedRAMP expects control responses at this level.

Within the `control-implementation` / `implemented-requirements` array, each entry includes:

- a required `uuid` field
- a required `control-id` field that cites the control using its id from the baseline.
- a required `statements` array. Each array entry includes:
  - a `statement-id` field that cites the control statement using its id from the baseline.
  - a `by-components` array
    - See Responding By Component for more information.

Multi-Part Statement Representation

```
system-security-plan:
  control-implementation:
    implemented-requirements:
```

```
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-1
      statements:
      - statement-id: ac-1_smt.a
        uuid: 11111111-2222-4000-8000-012000010100
        by-components:
          [content cut]
```

Non-Typical

If there are no lettered parts in the control definition, such as with AC-2 (1), there must be exactly one statement assembly.

Single-Statement Representation

AC-2(1) What is the solution and how is it implemented?

A single-statement representation is identical to a typical multi-part statement representation, except for the following:

- there is only one entry in the `statements` array
- the `statement-id` value cites the baseline ID for the `statement` part itself instead of one of its child parts.

```
system-security-plan:
  control-implementation:
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-2.1
      statements:
      - statement-id: ac-2.1_smt
        uuid: 11111111-2222-4000-8000-012000010100
        by-components:
          [content cut]
```

Control Response: Policies, Procedures, Plans, RoB, and Guides

Most FedRAMP-required attachments derive their requirement from one or more NIST SP 800-53 controls. With an OSCAL SSP, the attachment is linked directly from the control. This is how tools know which attachment satisfies each requirement.

Control ID	Artifact to Link	Expected
Each `-1`	**Policy**	1
Each `-1`	**Procedure(s)**	1+
SA-5 (`id`=`sa-5`)	**Appendix D: User Guide**	1
PL-4 (`id`=`pl-4`)	**Rules of Behavior**	1
CP-2 (`id`=`cp-2`)	**Information System Contingency Plan (ISCP)**	1
CM-9 (`id`=`cm-9`)	**Configuration Management Plan (CMP)**	1
IR-8 (`id`=`ir-8`)	**Incident Response Plan (IRP)**	1
CA-7 (`id`=`ca-7`)	**Continuous Monitoring Plan**	1
SR-2 (`id`=`sr-2`)	**Supply Chain Risk Management Plan (SCRMP)**	1

Retrofit MVP

For Retrofit MVP, simply use a `links` array in the `implemented-requirements` entry for each "-1" control.

```
system-security-plan:
  control-implementation:
    description: There is one control in this example. Follow this pattern for each
      additional control.
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000010000
      control-id: ac-1
      links:
      - href: ./AC_Policy.docx
        rel: policy
        media-type: application/docx
```

```
    - href: ./AC_Procedure.docx
      rel: procedure
      media-type: application/docx
```

Normalized

For Retrofit Advanced, and all New adoption:

- Attach the document as a back-matter resource.
- Create a component that represents the document
- Specify the component in the control response

Attach Document

Attach each document as `back-matter` / `resources` entries and include a `props` array with:

- `name` set to `type`
- `value` set to `policy`, `procedure`, `plan`, `users-guide` or `rules-of-behavior`


```
system-security-plan:

  back-matter:
    resources:
    - uuid: 11111111-2222-4000-8000-001000000005
      title: Access Control and Identity Management Policy
      description: A single policy that addresses both the AC and IA families.
      props:
      - name: type
        value: policy
      - name: published
        value: '2023-01-01T00:00:00Z'
      - name: version
        value: '1.2'
      rlinks:
      - href: ./attachments/policies/sample_AC_and_IA_policy.pdf
        media-type: application/pdf
```


Create Component

Create a component for each document in `system-implementation` / `components` and include:

- a `props` array with one entry:
  - `name` set to `implementation-point`
  - `value` set to `internal` if the document is system-specific; or
  - `value` set to `external` and `class` set to `corporate` if the document is Corporate
- a `links` array with one entry:
  - `href` contains a URI fragment that cites the back-matter resource
    - a hashtag (`#`) followed by the UUID of the back-matter resource.
  - `rel` contains `attachment`

All other fields depicted in the example are required by OSCAL to be present.


```yaml
system-security-plan:

  system-implementation:
    components:
    - uuid: 11111111-2222-4000-8000-009000600001
      type: policy
      title: Access Control and Identity Management Policy
      description: 'This is a corporate AC policy used for the system.'
      props:
      - name: implementation-point
        value: external
        class: corporate
      links:
      - href: '#11111111-2222-4000-8000-001000000005'
        rel: attachment
      status:
        state: operational
```


Control Response

Use `implemented-requirements` / `statements` / `by-components` entries in every control response that cites the document.


```yaml
system-security-plan:
```

```yaml
control-implementation:

  implemented-requirements:
  - uuid: 11111111-2222-4000-8000-012000010000
    control-id: ac-1

    statements:
    - statement-id: ac-1_smt.a
      uuid: 11111111-2222-4000-8000-012000010100

      by-components:
      - component-uuid: 11111111-2222-4000-8000-009000600001
        uuid: 11111111-2222-4000-8000-012000010102
        description: Describe how this policy satisfies part a.
        implementation-status:
          state: implemented
```

Inheritence and Customer Responsibilities

For systems that may be leveraged, OSCAL enables a robust mechanism for providing both inheritance details as well as customer responsibilities (referred to as consumer responsibilities by NIST). OSCAL is designed to enable leveraged and leveraging system SSP details to be linked by tools for validation.

Within the appropriate `by-component` assembly, include an export assembly. Use `provided` to identify a capability that may be inherited by a leveraging system. Use `responsibility` to identify a `customer responsibility`. If a `responsibility` must be satisfied to achieve inheritance, add the `provided-uuid` flag to the `responsibility` field.

Representation

```
system-security-plan:
  control-implementation:
    implemented-requirements:
    - uuid: 11111111-2222-4000-8000-012000020000
      control-id: ac-2

      statements:
      - statement-id: ac-2_smt.a
        uuid: 11111111-2222-4000-8000-012000020100
        by-components:
        - component-uuid: 11111111-2222-4000-8000-009000000000
          uuid: 11111111-2222-4000-8000-012000020102
          description: 'Confidential control response.'
          implementation-status:
            state: implemented

          export:
            provided:
            - uuid: 11111111-2222-4000-8000-015000000001
              description: This system's statement of capabilities which may be inherited
                by a customer's leveraging systems toward satisfaction of AC-2, part a.

            responsibilities:
            - uuid: 11111111-2222-4000-8000-016000000001
              provided-uuid: 11111111-2222-4000-8000-015000000001
```

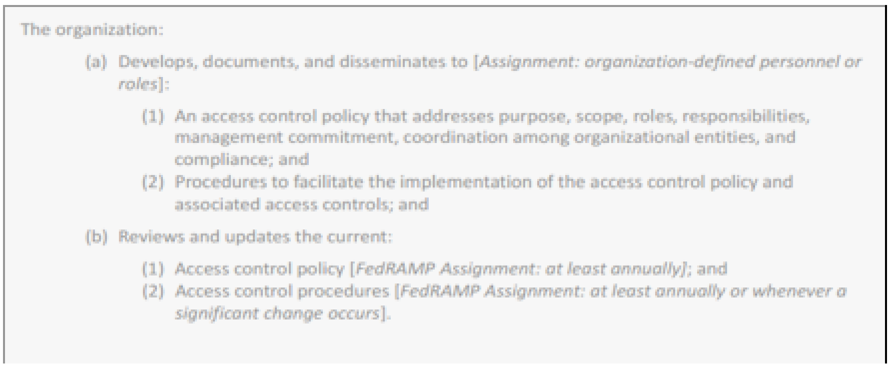
```
                description: 'Leveraged system''s statement of a leveraging system''s
                  responsibilities in satisfaction of AC-2, part a.'
                responsible-roles:
                - role-id: cloud-service-provider
                  party-uuids:
                  - 11111111-2222-4000-8000-004000000001
```

See the [NIST OSCAL Leveraged Authorization Presentation](#) for more information.

Leveraged Authorization Response: Inheriting Controls, Satisfying Responsibilities

When the current system is inheriting a control from or meeting customer responsibilities defined by an underlying authorization, the leveraged system must first be defined as described in the [*Response: Identifying Inheritable Controls and Customer Responsibilities*](#) section, and documented a `component` int the leveraging system SSP before it may be referenced in a control response. The `by-component` assembly references these components.

IMPORTANT: The leveraged system may provide a single `component` representing the entire leveraged system or may provide individual system components as well. In either case, the `inherited-uuid` property in the `component` must have the `value` flag set to the UUID of the leveraged system or component.


The organization:

(a) Develops, documents, and disseminates to [*Assignment: organization-defined personnel or roles*]:

(1) An access control policy that addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and

(2) Procedures to facilitate the implementation of the access control policy and associated access controls; and

(b) Reviews and updates the current:

(1) Access control policy [*FedRAMP Assignment: at least annually*]; and

(2) Access control procedures [*FedRAMP Assignment: at least annually or whenever a significant change occurs*].

Representation

```
system-security-plan:
```

```yaml
system-implementation:
  components:
  - uuid: 11111111-2222-4000-8000-009000100004
    type: system
    title: Leveraged Authorized System
    description: Briefly describe the leveraged system.
    status:
      state: operational


control-implementation:
  implemented-requirements:
    statements:
      by-components:
      - component-uuid: 11111111-2222-4000-8000-009000000004
        uuid: 11111111-2222-4000-8000-012000020104
        description: For the portion inherited from an underlying FedRAMP-authorized
          provider, describe **what** is inherited.
        implementation-status:
          state: implemented
        inherited:
        - uuid: 11111111-2222-4000-8000-017000000001
          provided-uuid: 11111111-0000-4000-9009-002001002001
          description: 'Optional description.'
        satisfied:
        - uuid: 11111111-2222-4000-8000-018000000001
          responsibility-uuid: 11111111-0000-4000-9009-002001002002
          description: 'Description of how the responsibility was satisfied.'
```


See the NIST OSCAL Leveraged Authorization Presentation for more information.

Citing Control Statements

OSCAL SSPs cite OSCAL baseline statement identifiers when representing control implementation responses. Citing the identifiers correctly is critical to machine processing.

Within OSCAL baselines, identifiers are assigned to `statement` parts and `item` parts for reference by SSPs.

The `statement` Part

All OSCal `parts` entries have:

- a required `id` field; and
- a required `name` field.

For every control in the FedRAMP baselines there is *exactly one* `parts` entry where `name` = `statement`. This is the `statement` part.

```
    - id: ac-2.1
      title: Automated System Account Management
      parts:
      - id: ac-2.1_smt
        name: statement
```

Simple Controls

For simple controls, the `statement` part has a `prose` field that includes the control requirement statement.

```
    - id: ac-2.1
      title: Automated System Account Management
      parts:
      - id: ac-2.1_smt
        name: statement
        prose: 'Support the management of system accounts using {{ insert: param, ac-
02.01_odp }}.'
```

The `id` value for the `statement` part (i.e. `ac-2.1_smt`) is cited by the SSP's `statements` array when responding to this control.

Controls with Child Statements

For a control with child statements (a., b., etc.), the `statement` part includes a nested `parts` array. Every element in the nested `parts` array has:

- a required `id` field; and
- a required `name` field. Always with a value of `item`.
- a `prose` field that includes this part of the control requirement statement.
- an additional nested `parts` array **IF** this part has child parts.

Each control in the FedRAMP OSCAL baselines has a `parts` array at the root of the control. Each `parts` entry includes:

- a required `id`
- a required `name`.

```
catalog:
  groups:
    controls:
    - id: ac-1
      title: Policy and Procedures
      parts:
      - id: ac-1_smt
        name: statement
        parts:
        - id: ac-1_smt.a
          name: item
          props:
          - name: label
            value: 'a.'
          prose: 'Develop, document, and disseminate to {{ insert: param, ac-1_prm_1 }}:'
```

> For SSP authoring, ignore any `parts` entry in the baseline outside of the `statement` part and its child parts. Other part types are for control assessments.

Response Point Properties

To aid SSP authoring tools in identifying the required statement level at which to respond, `response-point` properties are included in the FedRAMP baselines.

SSP authoring tools should limit the scope of `response-point` property searches to the `statement` part and its child parts. Ignore `response-point` properties in the parts related to assessments.

A `response-point` property appears in the `props` array and includes:

- a `name` set to `response-point`
- a `ns` set to `http://fedramp.gov/ns/oscal`
- a `value` with a value that is any string and can be ignored.

```
      - id: ac-2.1
        title: Automated System Account Management
        parts:
        - id: ac-2.1_smt
          name: statement
          props:
          - name: response-point
            ns: http://fedramp.gov/ns/oscal
            value: You must fill in this response point.
          prose: 'Support the management of system accounts using {{ insert: param, ac-
02.01_odp }}.'
```

When an SSP tool encounters a `parts` entry that contains this property, it should be presented to users of SSP authoring tools as the expected level of response for that control.