

Table of Contents

REACTR

Introduction & Overview

[Introduction](#)

[Release Notes](#)

RPS Patching in REACTR

[About RPS Patching in REACTR](#)

[How to Create, Edit, and Download RPS Patches in REACTR](#)

[RPS Patch Product Types](#)

[RPS Patch Conditions](#)

[The Baseline Patch List](#)

Instance Definitions

[Instance Definitions Technical Design](#)

[Instance Definitions User Guide](#)

REACTR

Last updated on June 28, 2021.

Last Reviewed and Approved on PENDING REVIEW

What is REACTR?

RPS Enterprise Automation, Configuration, and Testing Routines (REACTR) is a Rapid Provisioning System (**RPS**) application written to assist in automating the build of Patches, Instance Definitions, and ISO images.

REACTR Release Notes

Last updated on March 30, 2021.

Last Reviewed and Approved on PENDING REVIEW

What's New in 2.7.1 (March 15, 2021)

Added

- Instance Definitions - Provides the ability to create, manage, and export Instance Definitions.
- Content Management System (CMS) - Provides a way to upload, manage, and download any type of content.

Changed

- Instance Definitions Items - Provides the ability to create and manage Instance Definition Items.
 - Converted from ASP.NET MVC to Blazor.
- Lots of bug fixes and small improvements under the hood to the Packaging and other existing features in REACTR.

About RPS Patching in REACTR

What is REACTR?

RPS Enterprise Automation, Configuration, and Testing Routines (REACTR) is a Rapid Provisioning System (**RPS**) application written to assist in automating the build of patches, Instance Definitions, and ISO images.

REACTR provides features to create and maintain patches and ISO images for use and deployment in RPS. Using REACTR, users may create new patches and ISOs with custom files or pre-built files from the Microsoft Catalog such as Windows updates and/or other Microsoft product updates.

Once the patches or ISOs have been created in REACTR, they can then be downloaded and loaded into RPS.

What is REACTR Patching Documentation?

The RPS Patching in REACTR documentation provides instructions on how to create, edit, delete, and download patches and ISOs in REACTR.

RPS Patching in REACTR documentation is accessible from the left-hand menu.

What is RPS?

Rapid Provisioning System (RPS) is a flexible and powerful automation tool for managing software installation, updates, and configuration, and provides features that allows users to consume and deploy patches and ISOs that have been created in REACTR.

What is RPS Patching Documentation?

The RPS Patching documentation provides details about the RPS patching system, including how to properly operate and maintain RPS; and how to load, deploy, and check the status of patches and ISOs in RPS.

- To access *RPS v4.0.0 Patching Documentation*, please visit: [RPS v4.0.0 Patching](#)
or select "RPS Patching" from the top menu and select "v4.0.0" from the dropdown.
- To access *RPS v3.1.0 Packaging Documentation*, please visit: [RPS v3.1.0 Packaging](#)
or select "RPS Patching" from the top menu and select "v3.1.0" from the dropdown.

How to Create, Edit, and Download RPS Patches in REACTR

Last updated on May 3, 2021.

Last Reviewed and Approved on PENDING REVIEW

Introduction

This document provides description and use guidance on the RPS Patch features in REACTR.

Assumptions

1. You have access to the REACTR website at <https://reactr.azurewebsites.us/>

REACTR Patch List Page Fundamentals

Patch Terms and Definitions

- Patch File – Zip file patch utilized for installation of a program application that needs to be loaded on to an RPS system.
- Patch Manifest – A file that provides characteristics and properties of a patch file

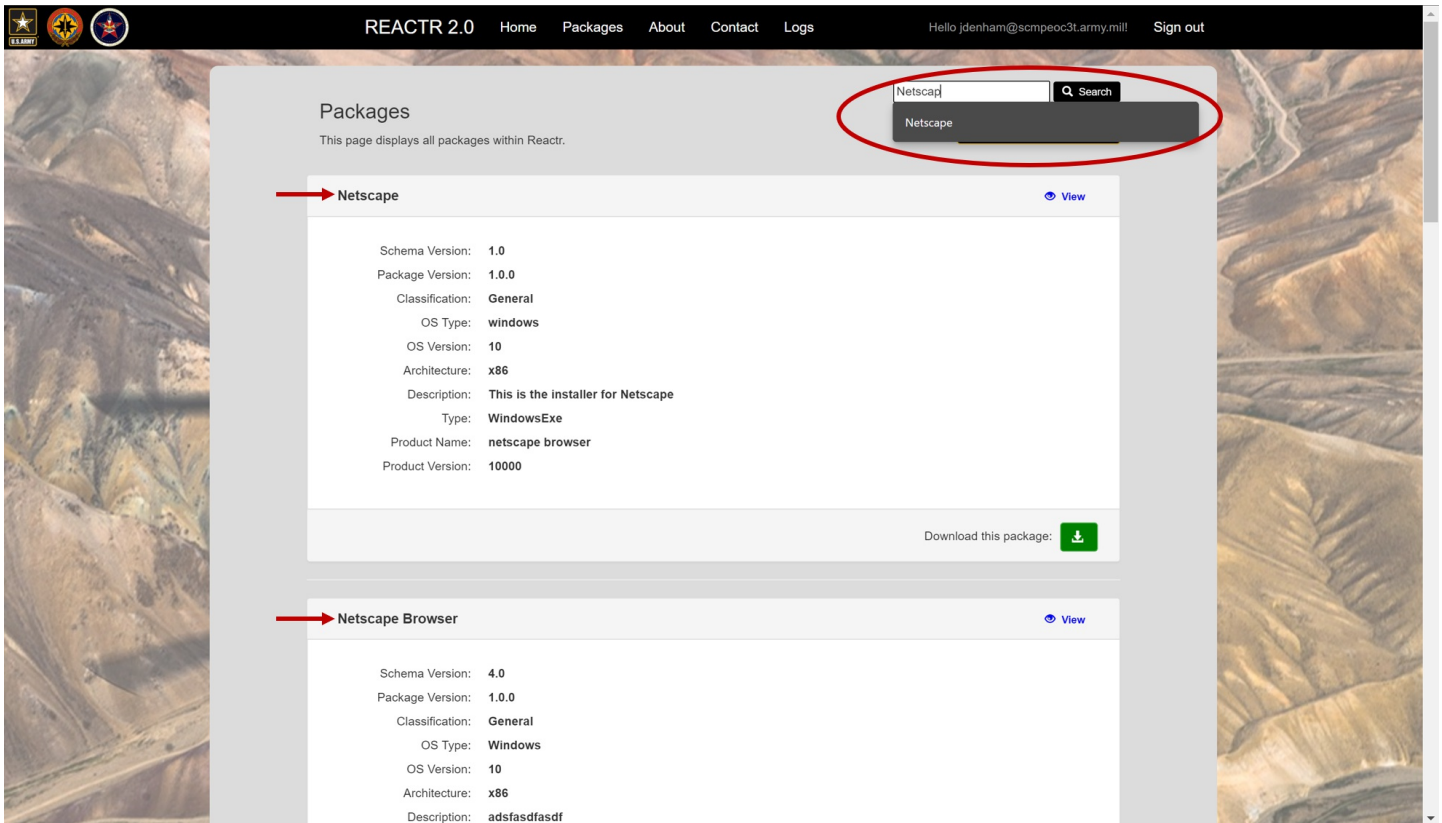
Patch Concepts

- REACTR is designed to simplify and automate the manual patch manifest building process by providing a form with all required fields, conditional requirements based on previous selections and providing dropdown menus for strings used in RPS that are contained in a manifest.

Search

Search the RPS patches on the REACTR site Patch List page is a useful way to find a desired patch. After finding your desired Patch through search, you can interact with the patch card as you would any other individual patch card. To perform a search:

1. Locate the **Search Bar** on the upper right side of the page.
2. Enter the term you would like to search for.
3. Search results will return with anything that includes your search term in the Patch Name.
4. Search results are limited to 25 per page. You may advance pages to find additional results.
5. For additional instructions, follow the guidance under [pagination](#).



Pagination

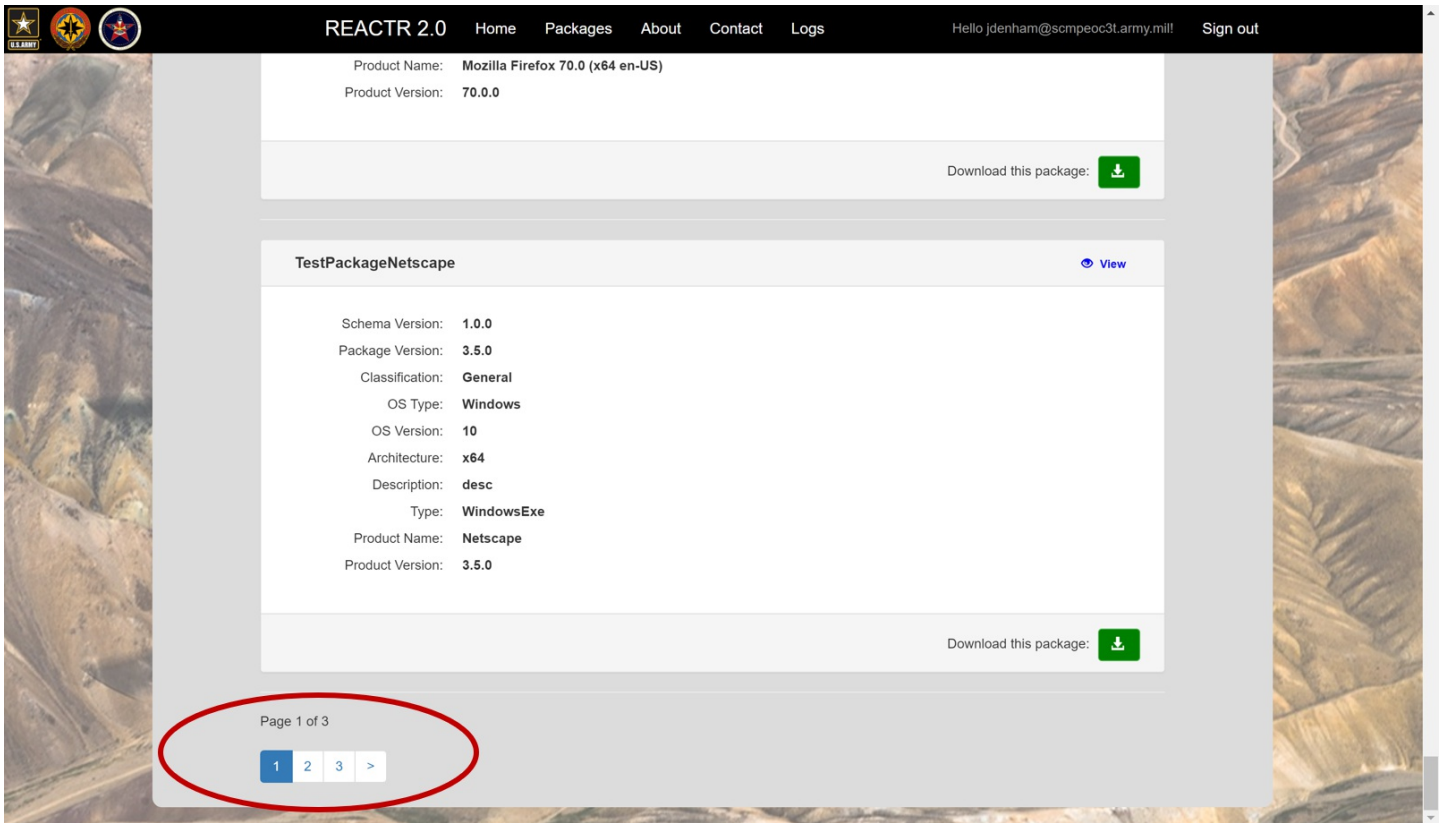
The REACTR Patch List page is designed to provide you with access to all RPS patches in the system. Each page will display up to a maximum of 25 individual patch cards.

1. To advance forward or backward a single page, you can click on the forward or backward arrow.

NOTE

If you are on the first page, the backward arrow will not show. Similarly, if you are on the last page, the forward arrow will not show.

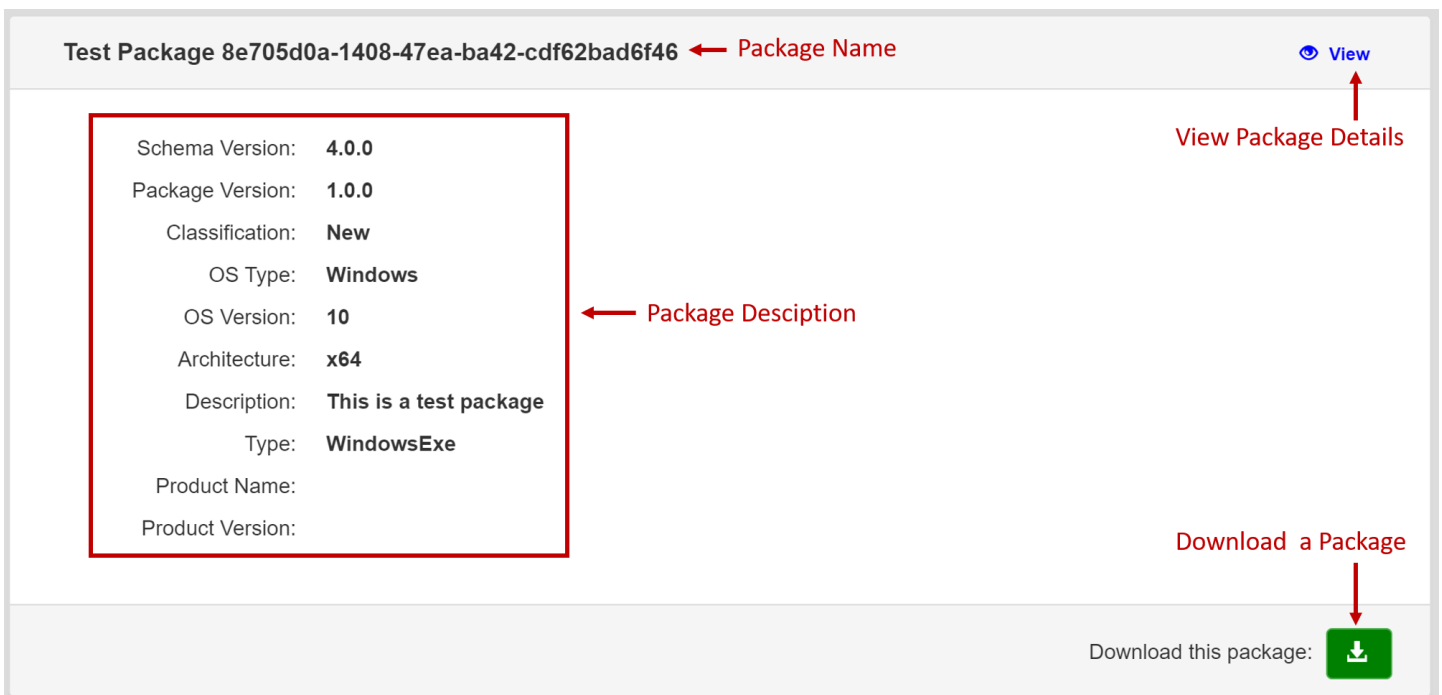
2. To advance to a specific page, you can click directly on the number page you would like to advance to.



Individual Patch Cards

The REACTR Patch List page is comprised of a series of Individual Patch Cards. An Individual Patch Card is a grouping of data points that provides identifying information and allows you to interact with a patch in several ways.

- **Patch Name:** This is the name of the patch. You can click on the **Patch Name** to view patch details, edit the patch, or delete the patch.
- **View:** Click on **View** to see the patch details. This is a view only screen.
- **Patch Description:** Displays patch pertinent information.
- **Download this Patch:** Click the green box to download the patch .rps zip file.
 - For detailed download instructions see the Download section of this document.



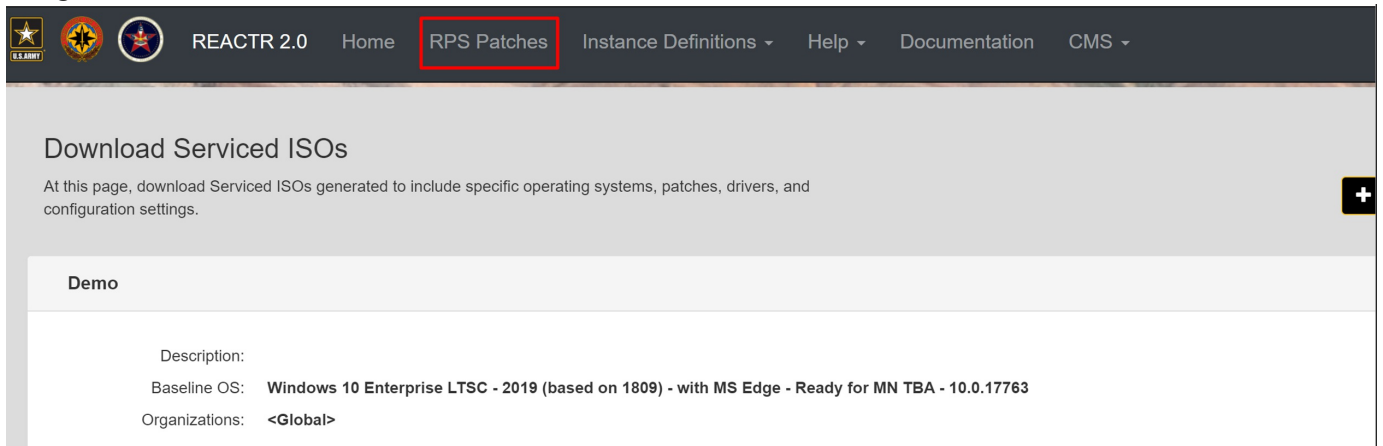
Create a Patch

WARNING

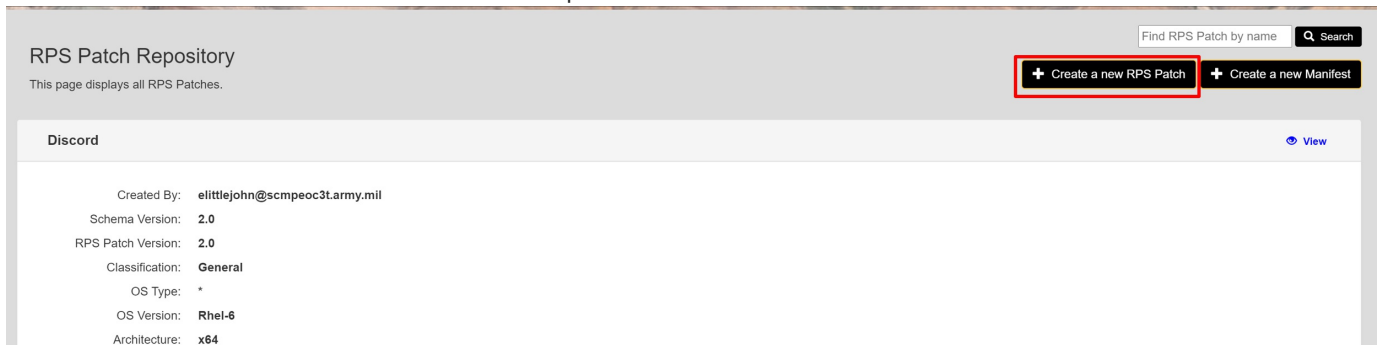
To create a patch, the user must have the role of "Patch Admin" or "Patch Creator".

To create a new patch:

1. Navigate to the RPS Patches screen.



2. Click the "Create a new RPS Patch" button at the top left of the screen.



3. Fill in the Patch Name (denoted as 'a' in the figure below).

NOTE

Only alphanumeric, hypens, and underscore characters are allowed in the Patch Name.

4. Fill in the Patch Version (denoted as 'b' in the figure below).

WARNING

Warning: The combination of Patch Name and Version must be unique.

NOTE

The version follows semantic versioning rules.

5. Selected the desired Schema Version (denoted as 'c' in the figure below).

NOTE

- o 1.0 - Indicates the patch will target RPS 3.1.
- o 2.0 - Indicates the patch will target RPS 3.1 and 4.0

6. Click the "Create" button (denoted as 'd' in the figure below).

Create a new RPS Patch

Patch Name: ?
a

Patch Version: ?
b

Manifest Schema Version: c ▾

d

7. The user will be navigated to the "Patch Edit" screen where they can begin to provide further information for the patch. See [Edit and View Form Fields Descriptions](#).

Edit a Patch

⚠ WARNING

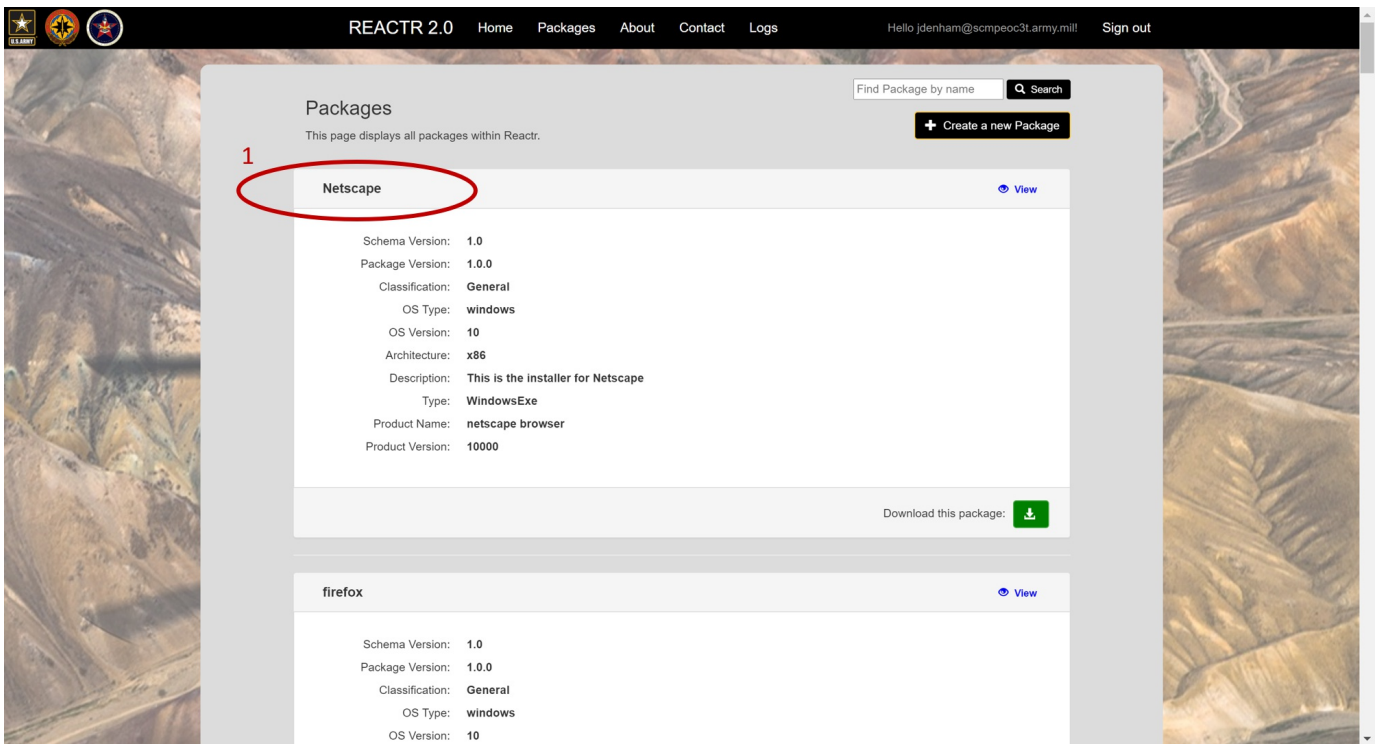
Warning: To create a patch, the user must have the role of "Patch Admin" or "Patch Creator".

ℹ NOTE

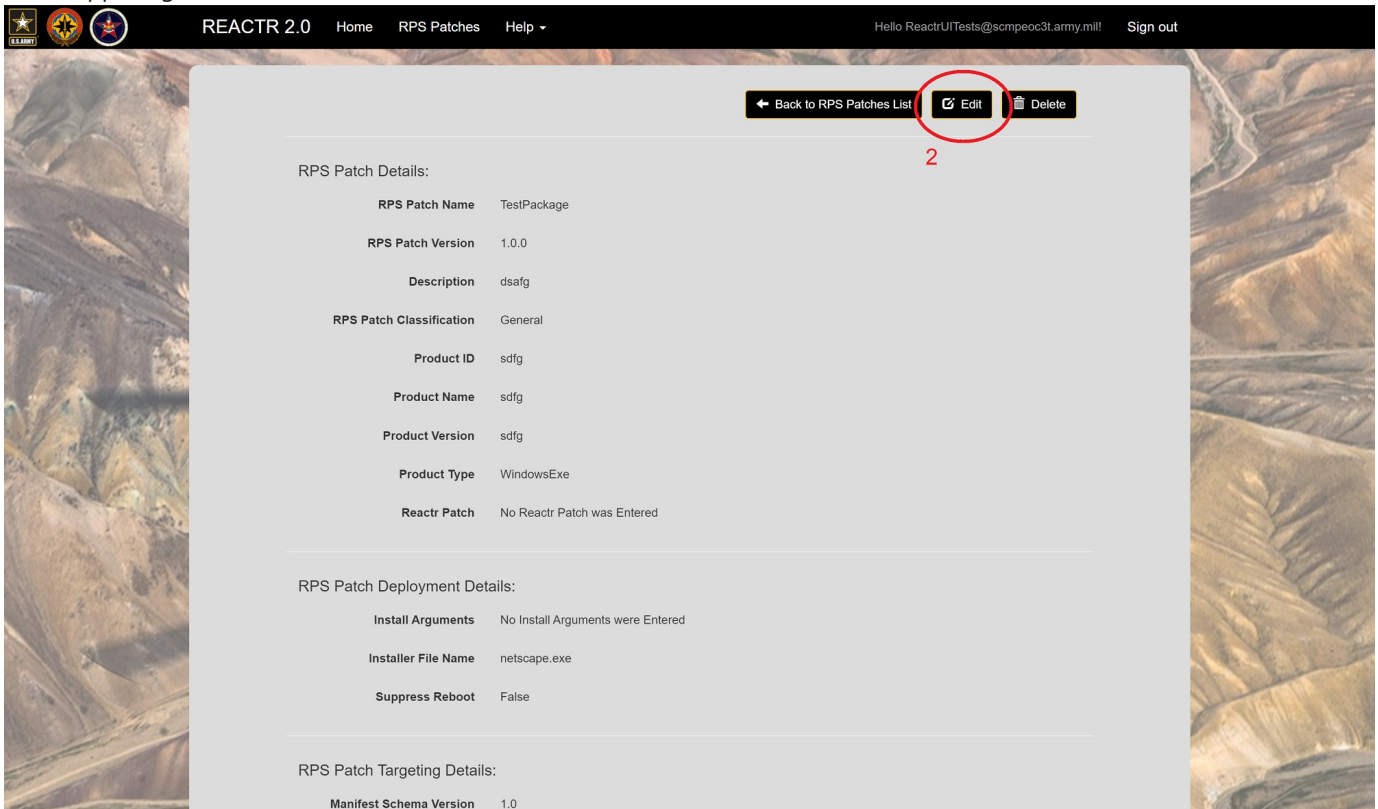
You CANNOT edit from the view page.

To modify an existing patch:

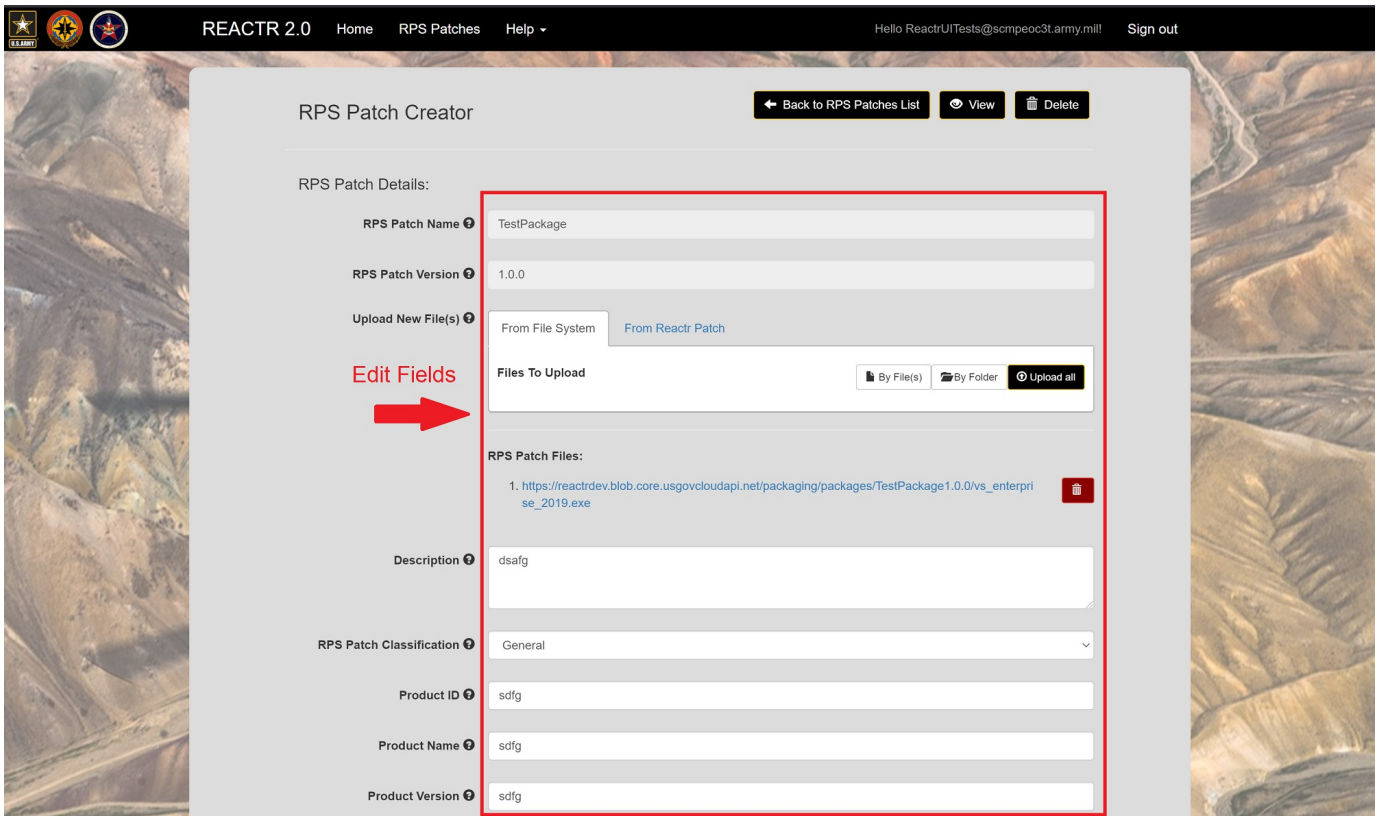
1. Click on the name of the patch you need to edit in the patch list.



2. In the upper right, click on Edit.

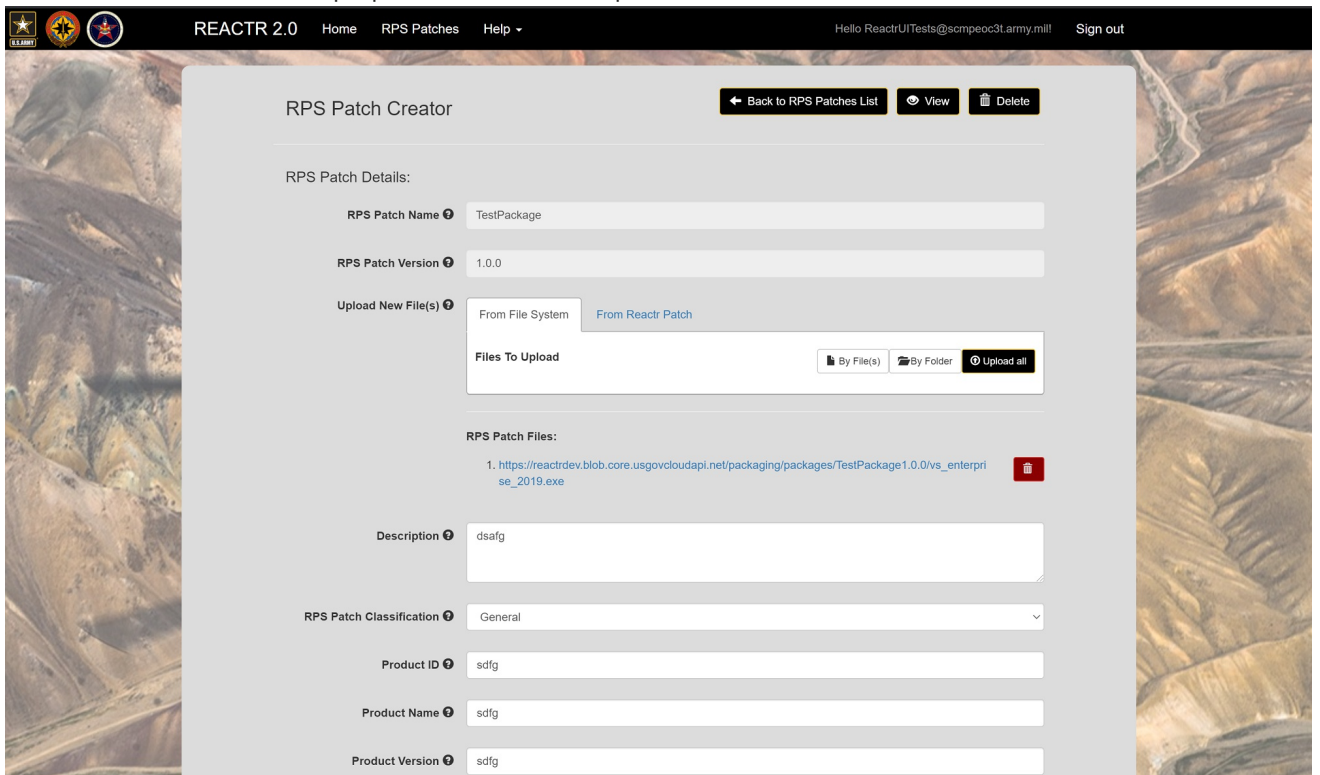


3. Fill in all required fields and any additional fields that may be required.



4. The edit form consists of 3 field types:

1. Free Text - Type in your desired data.
2. Drop Down - Select from pre-determined set of data.
3. Tabbed Selection – Select the proper tab to reveal the options in a scrollable field.



Edit and View Form Fields Descriptions

RPS Patch Details:

RPS Patch Name

- Name of the patch.

- The combination of Patch Name and Patch Version are used to uniquely name the Patch.
- If you want Patches to install in a certain order (chained together) then they must have the same Patch Name and incremented Patch Versions where the lowest versions are installed first.

⚠ IMPORTANT

This field is required.

ℹ NOTE

This value was set when the patch was created and cannot be changed.

RPS Patch Version

- Version of the patch.

⚠ IMPORTANT

This field is required.

ℹ NOTE

This value was set when the patch was created and cannot be changed.

ℹ NOTE

This is different than the version of the product that the patch contains. This is the version of the patch itself that the user is creating.

- The combination of Patch Name and Patch Version are used to uniquely name the Patch.
- The Patch Version be in a semantic version format (e.g. 70.1.2 or 1.10.11 or 9.8.1).
- If you want Patches to install in a certain order (chained together) then they must have the same Patch Name and incremented Patch Versions where the lowest versions are installed first.

Description

- Description of the patch – what the patch does, its purpose, etc.

⚠ IMPORTANT

This field is required.

Patch Classification

- Classification of the patch that helps to describe its purpose, its importance, etc.
 - For example, is it a new install? A script? A critical update? etc.

⚠ IMPORTANT

This field is required.

Product ID

- ID of the product that the patch contains.

⚠ WARNING

This field is required when the Product Type is: WindowsExe, WindowsMsi, WindowsCabinet, or WindowsHotfix

ℹ NOTE

This is important because it might be used to install/deploy/update/etc. the product inside the patch.

Product Name

- Name of the product that the patch contains.
 - This describes the primary product or item that the patch will be supporting/updating/installing/deploying/etc.

⚠ WARNING

This field is required when the Product Type is: WindowsExe, WindowsMsi, or LinuxRpm

Product Version

- Version of the product that the patch contains.

⚠ WARNING

This field is required when the Product Type is: WindowsExe or WindowsMsi

ℹ NOTE

This is different than the version of the patch since this is the version of the actual product that will be installed/deployed/updated/etc.

Product Type

- The type of product to be installed/deployed/updated/etc.

⚠ IMPORTANT

This field is required.

ℹ NOTE

This will determine how RPS deploys the product within the patch. For example, a Windows MSI will be deployed differently than a Windows EXE or a Linux RPM.

ℹ NOTE

After this value is set and saved the first time, this field becomes uneditable.

- For more information on Product Types see: [RPS Patch Product Types](#)

REACTR Patch

- This is the REACTR Patch that is associated with an RPS Patch.
- REACTR Patches are patches that REACTR automatically scrapes from the Microsoft Update Catalog and includes things such as Windows updates and other Microsoft product updates.
- Having the ability to select from an existing REACTR Patch makes it easier to create RPS Patches because the user won't

need to upload any files manually.

RPS Patch Deployment Details:

Install Arguments

- Additional arguments needed to install a patch.

Installer File Name

- The filename of the main installer within the patch. This is the file that will get executed to start the install/deploy/update process of the patch.

IMPORTANT

This field is required.

Suppress Reboot

- If set to 'True' then RPS will suppress the automatic reboot process after the patch is installed/deployed/updated/etc.

IMPORTANT

This field is required.

Targeting Details:

Manifest Schema Version

- The version of the patch manifest.

NOTE

Currently this version correlates with the Patch System Versions, but we might change this fact soon. The patching system might eventually have different versioning compared to Patch System Versioning.

Manifest Schema Version compatible With Patch System Versions From To

- The versions of RPS that this patch manifest schema version is compatible with. This gives a range of Patch System Versions that the patch is compatible with.

NOTE

Although this may change, currently the versions correlate with the Patch System Versions.

OS Type

- The OS Type that is compatible with the patch. For example: Windows, Linux, ESX, FreeBSD

IMPORTANT

This field is required.

OS Version

- The OS version that is compatible with the patch. For example: Windows-10, ESX-6.0

⚠ IMPORTANT

This field is required.

Architecture

- The architecture that is compatible with the patch. For example: x86, x64

⚠ IMPORTANT

This field is required.

Conditions

- Conditions are additional ways that a patch can be associated with targets in RPS.
 - If both a patch and 1 or more targets have the same condition then an assignment between the patch and target(s) will occur. This means that the targets must have properties that contain the condition(s) name and values.
 - Note: Conditions support full and partial wildcards (asterisk - *) using the LIKE or NOTLIKE operators.
 - Note: regular expressions are supporting with the MATCH or NOTMATCH operators.
- For more information on conditions see: [RPS Patch Conditions](#).

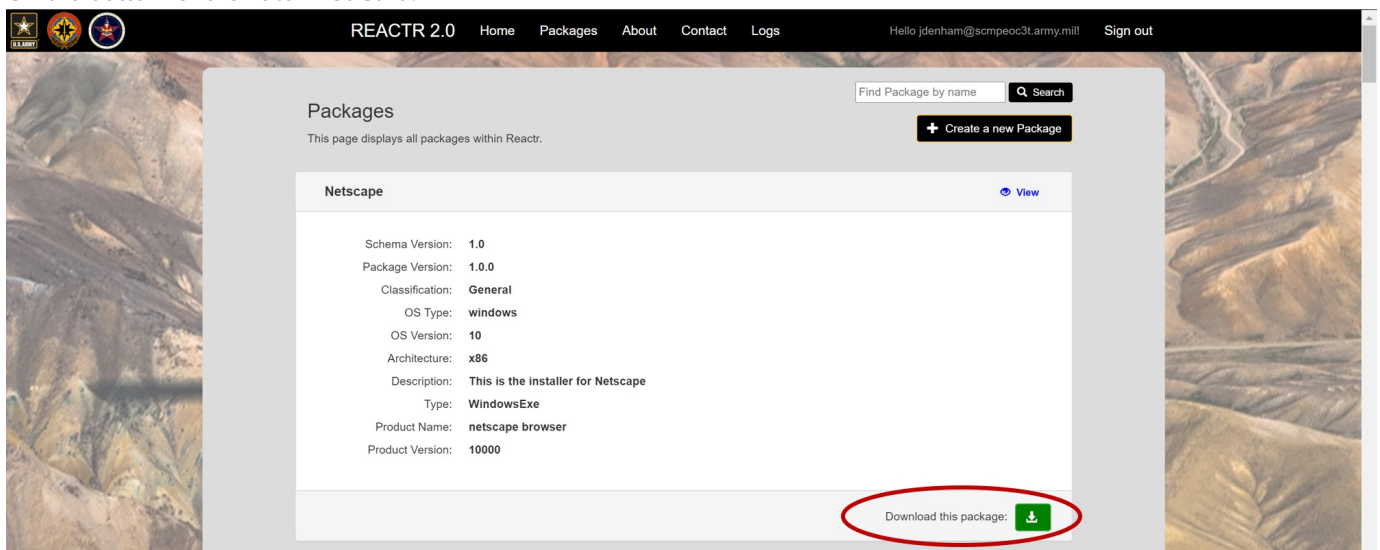
Patch File:

- The download of the actual RPS Patch that contains the patch's manifest, files that have been uploaded, and any REACTR Patch files.

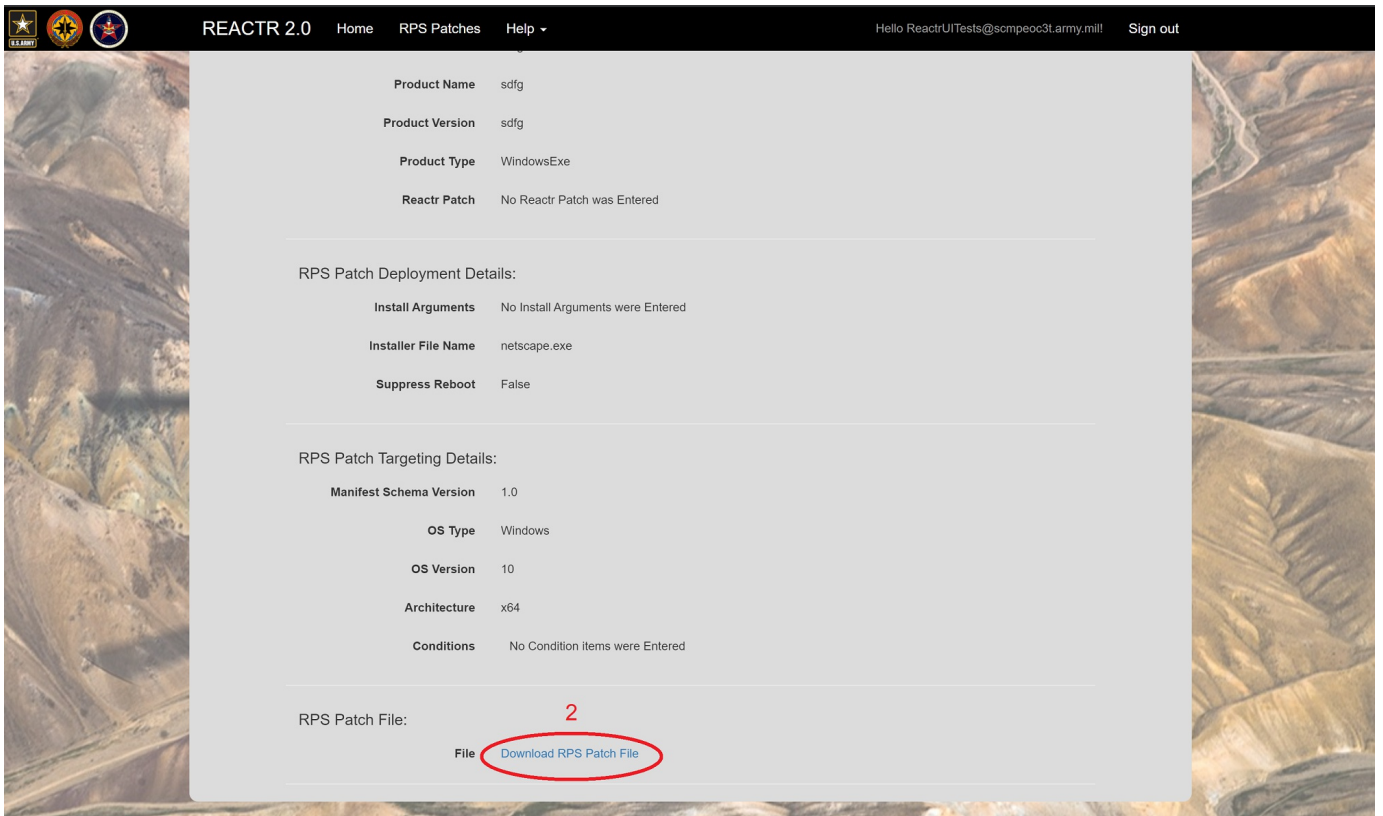
Download a Patch

Patches download as a .RPS zip file and are available to download for use in two locations:

1. On the bottom of the Patch List Card.

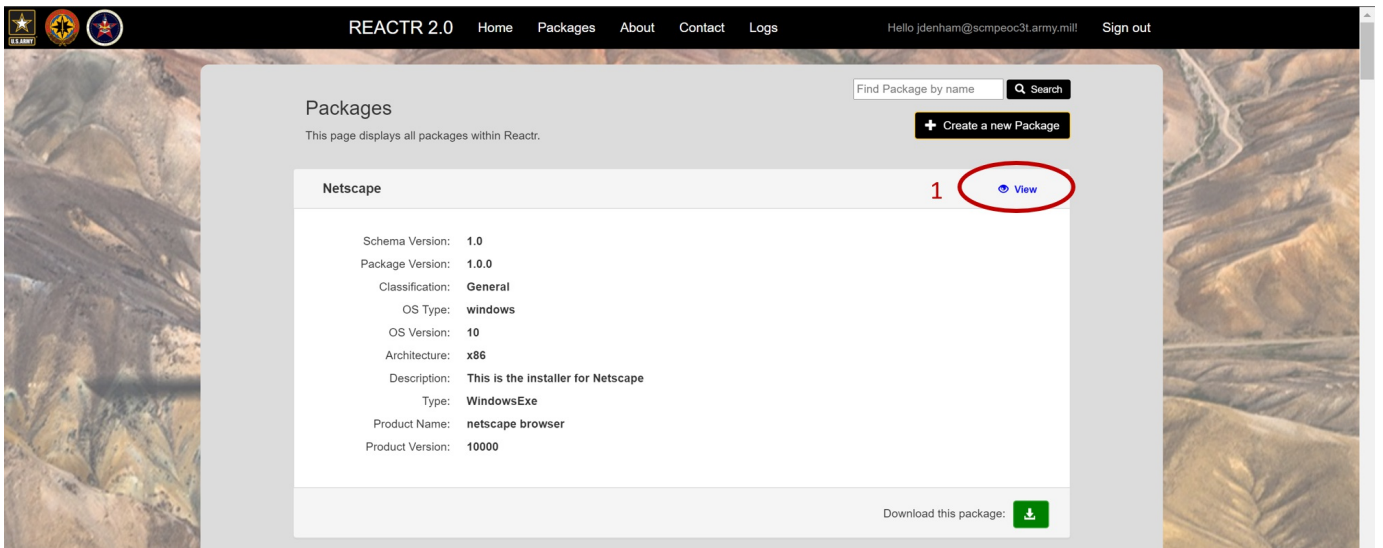
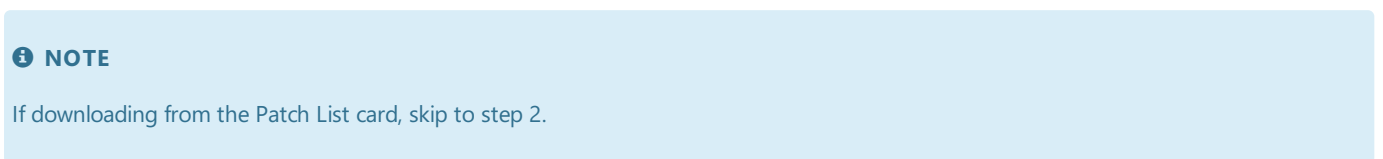


2. On the bottom of the View page.

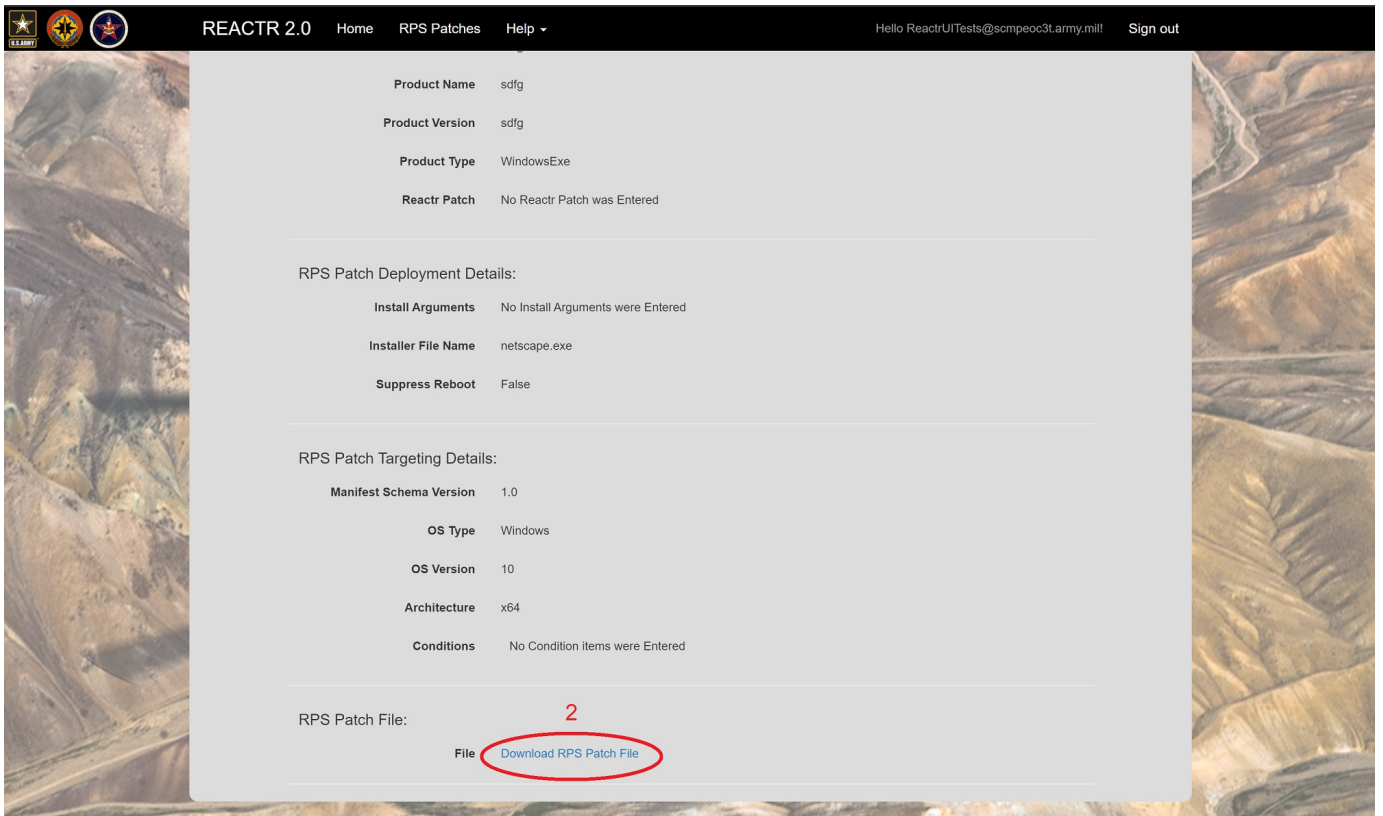


To download:

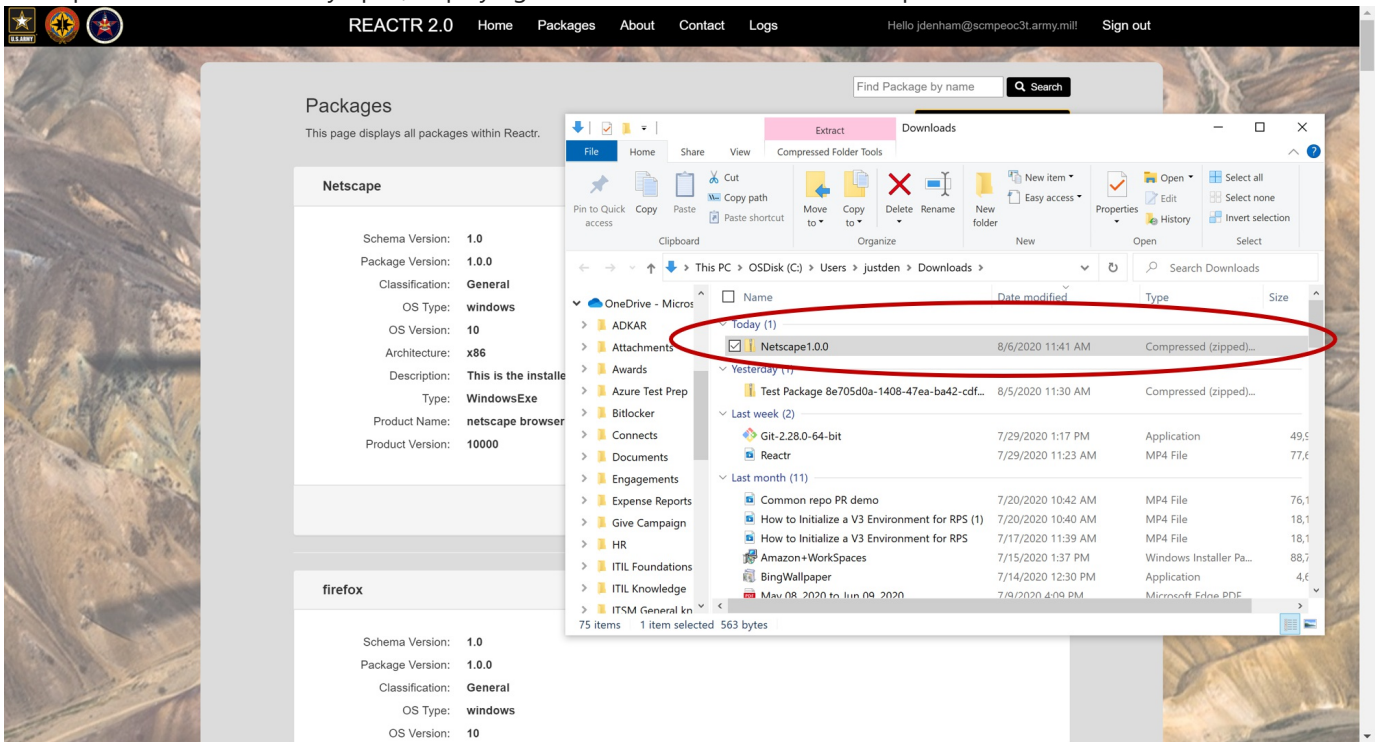
1. From the Patch List Page, Click on View.



2. Click on Download on either the Patch List Card or on the Patch Details View Page.



3. File Explorer will automatically open, displaying the downloaded location of the zip file.



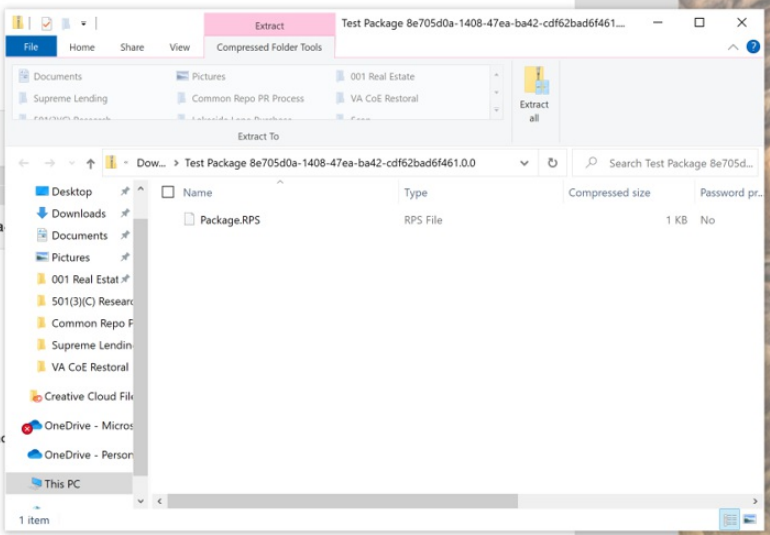
4. You may extract and use the zip file by double clicking to open it.



Description: **This is a test package**
Type: **WindowsExe**
Product Name:
Product Version:

Test Package 8e705d0a-1408-47ea-ba42-cdf62bad6f461...

Schema Version: **4.0.0**
Package Version: **1.0.0**
Classification: **New**
OS Type: **Windows**
OS Version: **10**
Architecture: **x64**
Description: **This is a test package**
Type: **WindowsExe**
Product Name:
Product Version:



Download this package:

RPS Patch Product Types

Last updated on June 21, 2021.

Last Reviewed and Approved on PENDING REVIEW

Introduction

This document provides guidance on how to get the values for Product ID, Product Name, and Product Version given the chosen Product Type.

Product Type is the type of product contained within a patch. The value chosen for Product Type will be used by RPS to determine how to deploy the product from the patch.

The Product Type must be accurate and match exactly with the product that will be deployed, otherwise deployment may not work as expected.

Certain fields are required to be filled in accurately depending on the chosen Product Type. This document describes which fields are required and how to obtain their values.

Product Types

Product Type: WindowsExe

The WindowsExe Product Type requires: Product ID, Product Name, and Product Version.

WindowsExe: Product ID

Product ID is required for WindowsExe.

To find the Product ID you need to install the product and look in the Windows Registry in the path:

`HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall` (You can use the "Find" command by right-clicking on the folder). The Product ID will be either the normal text name of the product or the key of the product, for example:

`{02A9F3AF-DB1E-30D4-B456-7B8BBB7BE2D3}`.

You can also use the `Get-WmiObject` command to get the Product ID, Product Name, and Product Version from an installed EXE.

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

NOTE
In the above command, IdentifyingNumber is equal to Product ID in REACTR.

WindowsExe: Product Name

Product Name is required for WindowsExe.

To find the Product Name you need to install the product and look in the Windows Registry in the path:

`HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`. The Product Name will be the value of the Display Name for the product, for example: `Microsoft ASP.NET Core 3.1.7 Shared Framework (x64)`.

You can also use the `Get-WmiObject` command to get the Product ID, Product Name, and Product Version from an installed EXE.

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

WindowsExe: Product Version

Product Version is required for WindowsExe.

To find the Product Version you need to install the product and look in the Windows Registry in the path:

HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall. The Product Version will be the value of the Display Version, for example: 3.1.7.0.

You can also use the `Get-WmiObject` command to get the Product ID, Product Name, and Product Version from an installed EXE.

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

Product Type: WindowsMsi

The WindowsMsi Product Type requires: Product ID, Product Name, and Product Version.

WindowsMsi: Product ID

Product ID is required for WindowsMsi. There are two ways to find the Product ID of a Windows MSI.

1. Install the product and look in the Windows Registry in the path:

HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall. The Product ID will be the key of the product, for example: {02A9F3AF-DB1E-30D4-B456-7B8BBB7BE2D3}.

2. Use a tool to get information directly from the MSI.

1. `Get-MsiProductInfo` - Use this PowerShell script to get the Product ID, Product Name, and Product Version:

1. Open PowerShell ISE As an Administrator
2. Open [Get-MsiProductInfo v2.ps1](#) script.
3. Replace the value of the \$Path variable with the path to your MSI installer.

```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Get-MsiProductInfo_v2.ps1* X
1 $Path = 'C:\MyInstaller.msi'
2
3 $sig = '@'
4 [DllImport("msi.dll", CharSet = CharSet.Unicode
5 private static extern UInt32 MsiOpenPackageW(s
6 [DllImport("msi.dll", CharSet = CharSet.Unicode
7 private static extern uint MsiCloseHandle(IntP
8 [DllImport("msi.dll", CharSet = CharSet.Unicode
9 private static extern uint MsiGetPropertyW(Int
10 private static string GetPackageProperty(strin
11 {
12     IntPtr MsiHandle = IntPtr.Zero;
```

NOTE

Make sure to save your changes to the script

4. Run the by clicking on the Run button or by pressing the 'F5' key.
2. Orca.exe - Use this Windows Installer SDK tool to get the Product ID, Product Name, and Product Version:
<https://docs.microsoft.com/en-us/windows/win32/msi/orca-exe>
3. Get-WmiObject - Use this command to get the Product ID, Product Name, and Product Version:

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

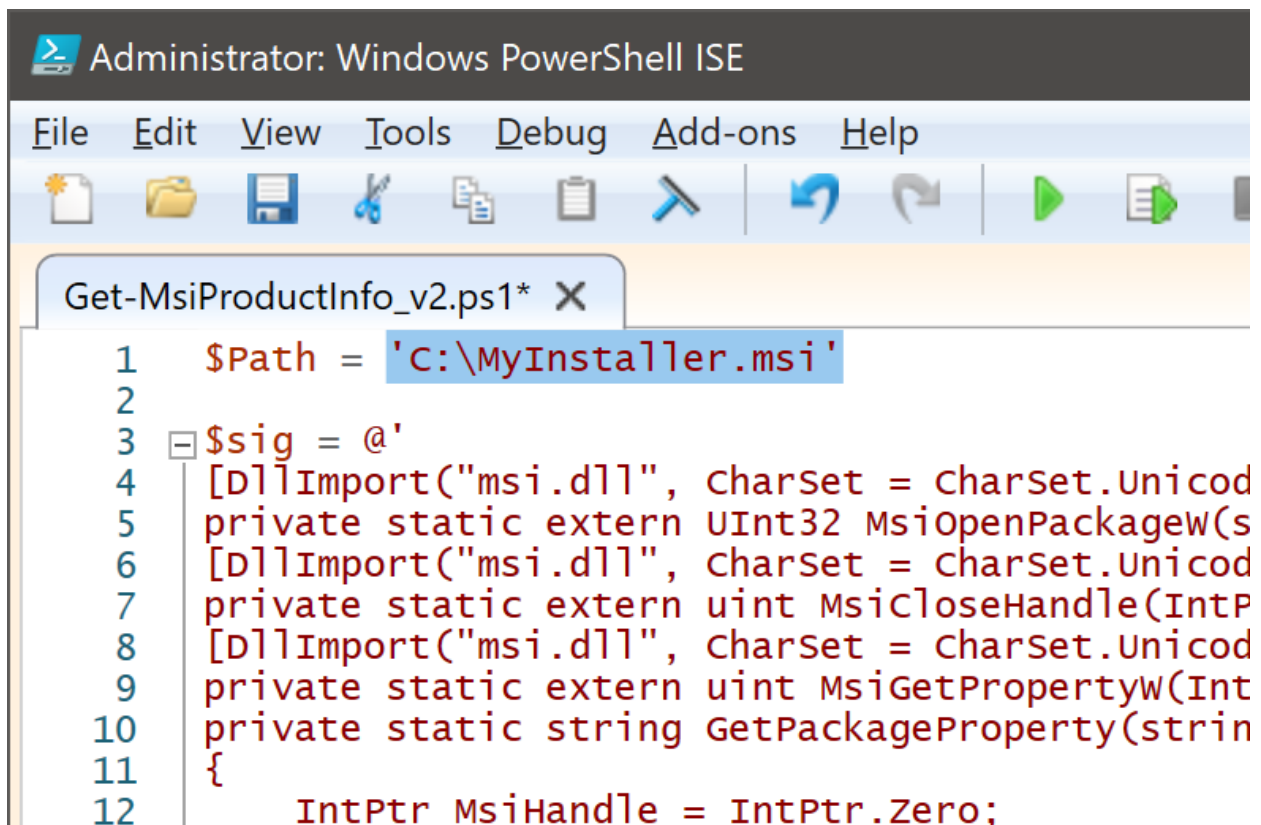
NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

WindowsMsi: Product Name

Product Name is required for WindowsMsi. There are two ways to find the Product Name of a Windows MSI.

1. Install the product and look in the Windows Registry in the path:
`HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`. The Product Name will be the value of the Display Name for the product, for example: `Microsoft ASP.NET Core 3.1.7 Shared Framework (x64)`.
2. Use a tool to get information directly from the MSI.
 1. Get-MsiProductInfo - Use this PowerShell script to get the Product ID, Product Name, and Product Version:
 1. Open PowerShell ISE As an Administrator
 2. Open [Get-MsiProductInfo_v2.ps1](#) script.
 3. Replace the value of the \$Path variable with the path to your MSI installer.



```
1 $Path = 'C:\MyInstaller.msi'
2
3 $sig = '@'
4 [DllImport("msi.dll", CharSet = CharSet.Unicode)
private static extern UInt32 MsiOpenPackageW(s
5 [DllImport("msi.dll", CharSet = CharSet.Unicode)
private static extern uint MsiCloseHandle(IntP
6 [DllImport("msi.dll", CharSet = CharSet.Unicode)
private static extern uint MsiGetPropertyW(Int
7 private static string GetPackageProperty(strin
8 {
9 IntPtr MsiHandle = IntPtr.Zero;
10
11
12
```

NOTE

Make sure to save your changes to the script

4. Run the by clicking on the Run button or by pressing the 'F5' key.
2. Orca.exe - Use this Windows Installer SDK tool to get the Product ID, Product Name, and Product Version:
<https://docs.microsoft.com/en-us/windows/win32/msi/orca-exe>
3. Get-WmiObject - Use this command to get the Product ID, Product Name, and Product Version:

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

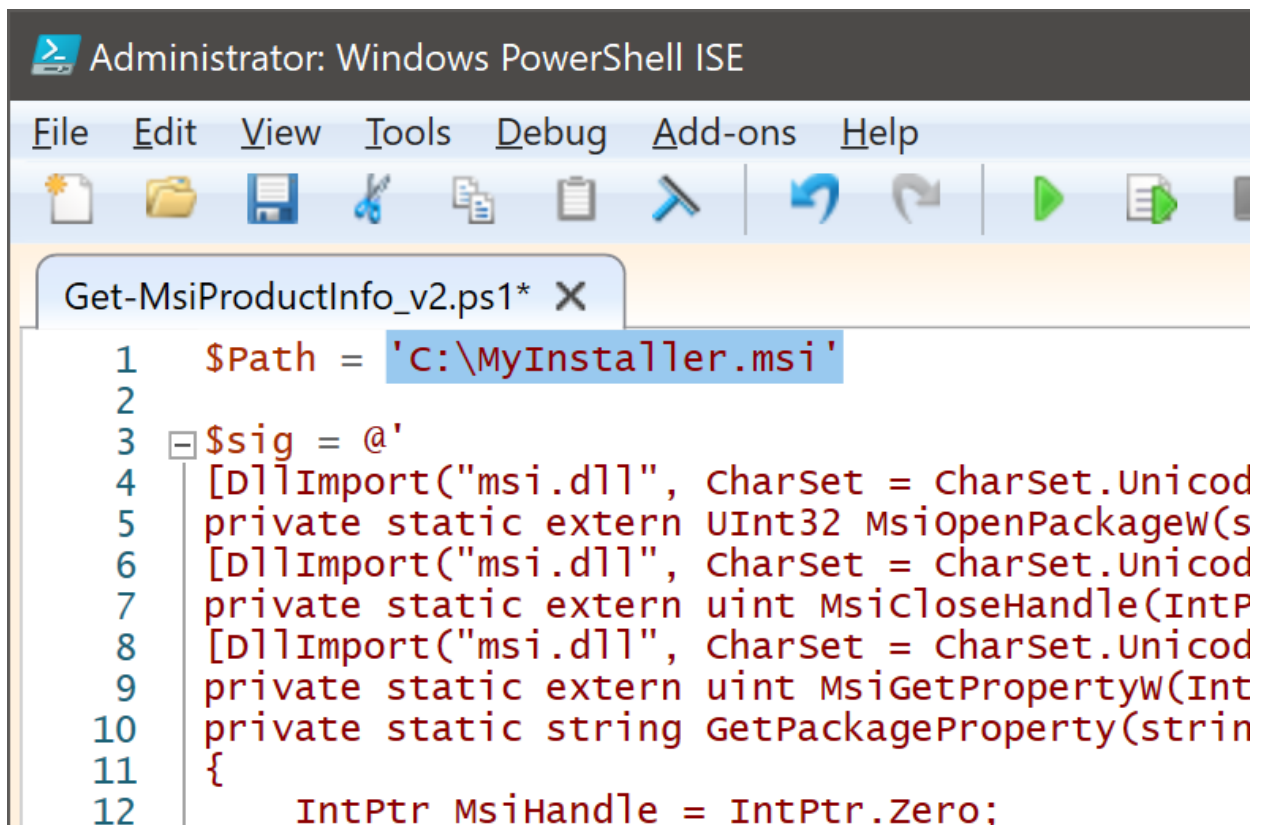
NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

WindowsMsi: Product Version

Product Version is required for WindowsMsi. There are two ways to find the Product Version of a Windows MSI.

1. Install the product and look in the Windows Registry in the path:
`HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`. The Product Version will be the value of the Display Version, for example: `3.1.7.0`.
2. Use a tool to get information directly from the MSI.
 1. Get-MsiProductInfo - Use this PowerShell script to get the Product ID, Product Name, and Product Version:
 1. Open PowerShell ISE As an Administrator
 2. Open [Get-MsiProductInfo_v2.ps1](#) script.
 3. Replace the value of the \$Path variable with the path to your MSI installer.



```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Get-MsiProductInfo_v2.ps1* X
1 $Path = 'C:\MyInstaller.msi'
2
3 $sig = @'
4 [DllImport("msi.dll", CharSet = CharSet.Unicode
5 private static extern UInt32 MsiOpenPackageW(s
6 [DllImport("msi.dll", CharSet = CharSet.Unicode
7 private static extern uint MsiCloseHandle(IntP
8 [DllImport("msi.dll", CharSet = CharSet.Unicode
9 private static extern uint MsiGetPropertyW(Int
10 private static string GetPackageProperty(strin
11 {
12     IntPtr MsiHandle = IntPtr.Zero;
```

NOTE

Make sure to save your changes to the script

4. Run the by clicking on the Run button or by pressing the 'F5' key.
2. Orca.exe - Use this Windows Installer SDK tool to get the Product ID, Product Name, and Product Version:
<https://docs.microsoft.com/en-us/windows/win32/msi/orca-exe>
3. Get-WmiObject - Use this command to get the Product ID, Product Name, and Product Version:

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

Product Type: WindowsHotfix

NOTE

WindowsHotfix displays as *WindowsMsu* in the Patch Product Type list when creating and editing a patch.

The WindowsHotfix Product Type requires: Product ID.

WindowsHotfix: Product ID

Product ID is required for WindowsHotfix.

To find the Product ID you can either look in the Microsoft Update Catalog or use the `Get-Hotfix` command. The Product ID of a Hotfix is the KB number of the product, for example: `KB3185911`.

Product Type: WindowsMsp

NOTE

The WindowsMsp Product Type is only available when creating an RPS Patch for Manifest Schema Version 2.0, which is compatible with **RPS 4.0**.

To install a WindowsMSP on **RPS 3.1** you may use the Script Framework by following this example: [Installing MSP Using the RPS Patching Script Framework](#).

The WindowsMsp Product Type requires: Product ID, Product Name, and Product Version.

WindowsMsp: Product ID

Product ID is required for WindowsMsp. There are two ways to find the Product ID of a Windows MSP.

1. Install the product and look in the Windows Registry in the path:

`HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`. The Product ID will be the key of the product, for example: `{02A9F3AF-DB1E-30D4-B456-7B8BBB7BE2D3}`.

2. Use a tool to get information directly from the MSP.

1. Orca.exe - Use this Windows Installer SDK tool to get the Product ID, Product Name, and Product Version:

<https://docs.microsoft.com/en-us/windows/win32/msi/orca-exe>

2. Get-WmiObject - Use this command to get the Product ID, Product Name, and Product Version:

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

WindowsMsp: Product Name

Product Name is required for WindowsMsp. There are two ways to find the Product Name of a Windows MSP.

1. Install the product and look in the Windows Registry in the path:

`HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`. The Product Name will be the value of the Display Name for the product, for example: `Microsoft ASP.NET Core 3.1.7 Shared Framework (x64)`.

2. Use a tool to get information directly from the MSP.

1. Orca.exe - Use this Windows Installer SDK tool to get the Product ID, Product Name, and Product Version:

<https://docs.microsoft.com/en-us/windows/win32/msi/orca-exe>

2. Get-WmiObject - Use this command to get the Product ID, Product Name, and Product Version:

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

WindowsMsp: Product Version

Product Version is required for WindowsMsp. There are two ways to find the Product Version of a Windows MSP.

1. Install the product and look in the Windows Registry in the path:

`HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`. The Product Version will be the value of the Display Version, for example: `3.1.7.0`.

2. Use a tool to get information directly from the MSP.

1. Orca.exe - Use this Windows Installer SDK tool to get the Product ID, Product Name, and Product Version:

<https://docs.microsoft.com/en-us/windows/win32/msi/orca-exe>

2. Get-WmiObject - Use this command to get the Product ID, Product Name, and Product Version:

```
Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name, Version
```

NOTE

In the above command, IdentifyingNumber is equal to Product ID in REACTR.

Product Type: WindowsCabinet

NOTE

RPS Patches using the WindowsCabinet Product Type are only compatible with Windows KB (Knowledge Base) updates that have the CAB file format. Other CAB files may not work using the WindowsCabinet product type.

You can test if a CAB file is compatible with the WindowsCabinet Product Type by using the `dism` command described in the [RPS Patch Provider](#) article.

If they are not compatible, then they may be able to be installed using the Script Framework Product Type and a custom PowerShell script that can install the file. For more information, see the [RPS Patching Script Framework](#) article.

NOTE

WindowsCabinet displays as *WindowsCab* in the Patch Product Type list when creating and editing a patch.

The WindowsCabinet Product Type requires: Product ID.

WindowsCabinet: Product ID

Product ID is required for WindowsCabinet.

To find the Product ID you can either look in the Microsoft Update Catalog or use the `Get-Hotfix` command. The Product ID of a Windows Cabinet is the KB number of the product, for example: `KB3185911`.

```
Get-Hotfix
```

Product Type: LinuxRpm

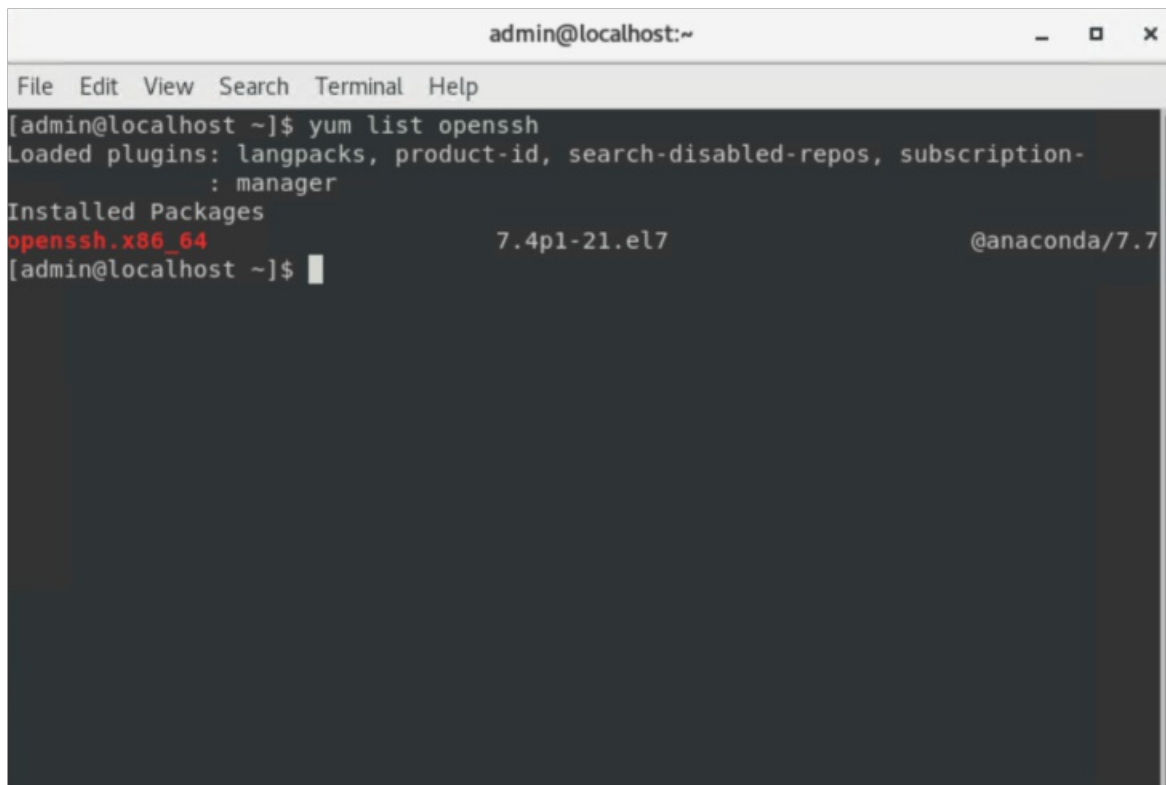
The LinuxRpm Product Type requires: Product Name.

LinuxRpm: Product Name

Product Name is required for LinuxRpm.

To find the Product Name you can use the command line yum utility. For example, to search for a patch called "openssh" you would use the command: `yum list openssh`.

From the yum search results, the Product Name is the name of the full yum patch name on the left-most side of the search results. (e.g. openssh.i386)

A terminal window titled 'admin@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'yum list openssh' and its output. The output includes 'Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager', 'Installed Packages', and a table with one entry: 'openssh.x86_64' (in red), '7.4p1-21.el7', and '@anaconda/7.7'. The prompt '[admin@localhost ~]\$' is visible at the end.

```
admin@localhost:~  
File Edit View Search Terminal Help  
[admin@localhost ~]$ yum list openssh  
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-  
                : manager  
Installed Packages  
openssh.x86_64                7.4p1-21.el7                @anaconda/7.7  
[admin@localhost ~]$
```

Product Type: Script Framework

The Script Framework Product Type does not require any specific Product-related fields (Product ID, Product Name, Product Version).

RPS Patch Conditions

Last updated on April 30, 2021.

Last Reviewed and Approved on PENDING REVIEW

Introduction

This document provides guidance on how to use conditions to provide RPS with more information to make more precise assignments between patches and targets. Any conditions that are added to a patch will be used to determine which targets the patch gets assigned to. Therefore, any conditions must match up with a target's properties (name and value) that correlate with each condition property name and value. In addition, the condition operator will determine how the comparison is made between the target and the patch.

Contains

The contains condition will check if a target property value does not contain a condition value within it. This comparison is case sensitive.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
ABC123	ABC123	true
ABC123	456	false
ABC123	BC12	true

The patch manifest will store this operator in the operator tag as: *like*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>like</Operator>
    <Value>MyCompu</Value>
  </PackageAssignmentCondition>
</Conditions>
```

support full and partial wildcards (asterisk - *) using the LIKE or NOTLIKE operators

Does not contain

The does not contain condition will check if a target property value does not contain a condition value within it. This comparison is case sensitive.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
ABC123	ABC123	false

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
ABC123	456	true
ABC123	BC12	false

The patch manifest will store this operator in the operator tag as: *notlike*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>notlike</Operator>
    <Value>MyCompu</Value>
  </PackageAssignmentCondition>
</Conditions>
```

support full and partial wildcards (asterisk - *) using the LIKE or NOTLIKE operators

Equal to

The equal to condition will check if a target property value equals a condition value by doing a case insensitive comparison.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
ABC123	ABC123	true
ABC123	456	false
ABC123	BC12	false

The patch manifest will store this operator in the operator tag as: *eq*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>eq</Operator>
    <Value>MyComputerName</Value>
  </PackageAssignmentCondition>
</Conditions>
```

Greater than

The greater than condition will check if a condition value is greater than a target property value.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
500	49	false

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
500	500	false
500	501	true

The patch manifest will store this operator in the operator tag as: *gt*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>NumberOfFolders</Property>
    <Operator>gt</Operator>
    <Value>500</Value>
  </PackageAssignmentCondition>
</Conditions>
```

Greater than or equal to

The greater than or equal to condition will check if a condition value is greater than or equal to a target property value.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
500	49	false
500	500	true
500	501	true

The patch manifest will store this operator in the operator tag as: *ge*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>NumberOfFolders</Property>
    <Operator>ge</Operator>
    <Value>500</Value>
  </PackageAssignmentCondition>
</Conditions>
```

Less than

The less than condition will check if a condition value is less than a target property value.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
500	49	true
500	500	false

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
500	501	false

The patch manifest will store this operator in the operator tag as: *lt*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>NumberOfFolders</Property>
    <Operator>lt</Operator>
    <Value>500</Value>
  </PackageAssignmentCondition>
</Conditions>
```

Less than or equal to

The less than or equal to condition will check if a condition value is less than or equal to a target property value.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
500	49	true
500	500	true
500	501	false

The patch manifest will store this operator in the operator tag as: *le*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>NumberOfFolders</Property>
    <Operator>le</Operator>
    <Value>500</Value>
  </PackageAssignmentCondition>
</Conditions>
```

Not equal to

The not equal to condition will check if a target property value is not equal to a condition value by doing a case insensitive comparison.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
ABC123	ABC123	false
ABC123	456	true

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
ABC123	BC12	true

The patch manifest will store this operator in the operator tag as: *ne*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>ne</Operator>
    <Value>MyComputerName</Value>
  </PackageAssignmentCondition>
</Conditions>
```

Regular expression does not match

The regular expression does not match condition will check if a target property value does not match a condition value using a regular expression.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
NFA	^(NFA)?(WNM)?(WNMA)?\$	false
WNM	^(NFA)?(WNM)?(WNMA)?\$	false
NFA123	^(NFA)?(WNM)?(WNMA)?\$	true

The patch manifest will store this operator in the operator tag as: *notmatch*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>notmatch</Operator>
    <Value>^(NFA)?(WNM)?(WNMA)?$</Value>
  </PackageAssignmentCondition>
</Conditions>
```

Regular expression match

The regular expression match condition will check if a target property value matches a condition value using a regular expression.

Examples:

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
NFA	^(NFA)?(WNM)?(WNMA)?\$	true
WNM	^(NFA)?(WNM)?(WNMA)?\$	true

TARGET PROPERTY VALUE	CONDITION VALUE	RESULT
NFA123	^(NFA)?(WNM)?(WNMA)?\$	false

The patch manifest will store this operator in the operator tag as: *match*.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>match</Operator>
    <Value>^(NFA)?(WNM)?(WNMA)?$</Value>
  </PackageAssignmentCondition>
</Conditions>
```

More Information on Regular Expressions

Match a value in a list of values (Logical OR)

The match or notmatch operators can be used to match multiple values where each value in the Value field wrapped in parentheses () and with a trailing question mark ? .

For an **exact** Value match, the full string in the Value field must be enclosed with a caret ^ and a dollar sign \$.

Manifest XML example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>Match</Operator>
    <Value>^(NFA)?(WNM)?(WNMA)?$</Value>
  </PackageAssignmentCondition>
</Conditions>
```

In this particular example, a Target with a Property of ComputerName will be assigned if its Value contains NFA, WNM, **and** WNMA.

Match all values in a list of values (Logical AND)

The match or notmatch operators can be used to match all values in a list where each value is prefixed with ?=.* and wrapped in parentheses (). Also, the entire list is prefixed with a caret ^ and suffixed with .*\$

Manifest XML Example:

```
<Conditions>
  <PackageAssignmentCondition>
    <Property>ComputerName</Property>
    <Operator>Match</Operator>
    <Value>^(?=.*The)(?=.*Best)(?=.*Computer).*$</Value>
  </PackageAssignmentCondition>
</Conditions>
```

In this particular example, a Target with a Property of ComputerName will be assigned if its Value contains **all** of the values: The, Best, and Computer.

For example:

TARGET PROPERTY VALUE	RESULT
The Best Computer	true
The Computer	false
The Best Computer Ever	true
Computer Best The	true
Best Computer	false

Common Regular Expressions

The following is a list of example regular expressions that may be used in patch conditions.

EXPRESSION	DESCRIPTION	EXAMPLES
<code>^(NFA)?(WNM)?(WNMA)?\$</code>	Case sensitive logical OR with exact match of any value	NFA, WNM, or WNMA . Then there is a match. If the value is: NF, or WN. Then there is not a match
<code>^(?i)(NFA)?(WNM)?(WNMA)?\$</code>	Case insensitive logical OR with exact match of any value	If the value is: nfa, wnm, or wnma . Then there is a match. If the value is: nf . Then there is not a match.
<code>^(?= .The)(?= .Best)(?= .Computer).\$</code>	Case sensitive logical AND with full match of all values	If the value is: The Best Computer . Then there is a match. If the value is: the best computer . Then there is not a match.
<code>^(?i)(?= .The)(?= .Best)(?= .Computer).\$</code>	Case insensitive logical AND with full match of all values	If the value is: the best computer . Then there is a match. If the value is: the worst computer . Then there is not a match.
<code>^[a-zA-Z0-9]*\$</code>	Alphanumerics with space allowed	If the value is: this is a 123 test . Then there is a match
<code>^[a-zA-Z0-9]*\$</code>	Alphanumerics without space allowed	If the value is: this is a 123 test . Then there is not a match

The Baseline Patch List

Last updated on January 19, 2021.

Last Reviewed and Approved on PENDING REVIEW

Introduction

This document outlines where REACTR gets its patches and which patches it chooses to download.

Microsoft Catalog

Patches for each of the baselines come directly from the Microsoft Catalog. REACTR automatically checks for patches from the Microsoft Catalog on a nightly basis and downloads any new patch that do not currently exist in the system.

There is currently no limit to the amount of **new** patches REACTR will download.

Supported Baselines

Patches are pulled from the Microsoft Catalog for the following baselines:

- Windows 10
- Windows Server 2019
- Windows 8.1

Patches

The following patch categories are included in the nightly REACTR search:

- Servicing Stack Updates
- Security Updates
- .NET Framework Update(s)
- Latest version of Adobe

Exclusions

Not every patch in the Microsoft Catalog are included in the nightly download. Preview patches and patches that are older than 1 year are ignored.

Instance Definitions Technical Design

Last updated on Sep 16, 2021.

Last Reviewed and Approved on PENDING REVIEW

Table of Contents

- [Instance Definitions Technical Design](#)
 - [Table of Contents](#)
 - [Introduction](#)
 - [Definition of Instance Definitions](#)
 - [Purpose of Generating Instance Definitions](#)
 - [Solution for Generating Instance Definitions](#)
 - [Overview](#)
 - [System Workflow](#)
 - [Technologies](#)
 - [Architecture](#)
 - [Modified Software](#)
 - [New Infrastructure](#)
 - [Security](#)
 - [Overview](#)
 - [Persistent Data Design](#)
 - [Data Technical Solution](#)
 - [Data Dictionary](#)
- [Glossary](#)
- [External Assets](#)

Introduction

This document describes REACTR's instance definitions feature and its technical design. The technical design was established based on the need for ways to simplify and improve the provisioning process.

Definition of Instance Definitions

Instance definitions are an abstraction layer on top of existing RPS types. Types currently allow developers define a specific type of resource or target item, its expected or possible Properties and their associated value types (string/int/etc.).

Instance definitions will allow developers to take these generic types and combine them together to create complex, nested entities composed of many different types through parent/child and assignment relationships, and the known default values for each individual Types' properties.

Examples: SNEs, TCNs, TSI-Large, laptop, could be a VM (because it could be a collection of components).

There are many existing types such as; Vehicle, Computer, VirtualMachine, NetworkConfiguration, etc. The entities soldiers and developers are now interacting with are complex and composed of many of these types in parent/child hierarchies.

Instance definitions are collections of type definitions that have the ability to create an object.

Type definitions are parts with detailed properties. The type definitions do not create objects, but they are able to define something you can create which is what an object will consist of.

Example: Virtual Machine, NIC, drive, etc.

Purpose of Generating Instance Definitions

The provisioning process for a Soldier or developer should be limited to selection of a template (instance definition) to use for provisioning and providing unit specific data files and RPS should be able to successfully provision with just this information supplied. The goal is to limit the number of choices and decisions that a soldier, or LSI, would need to make during provisioning.

By simplifying the provisioning process and removing the number of choices a soldier, or LSI, needs to make in the field, RPS will reduce the likelihood of human error and increase likelihood of adoption due to the improved ease of use.

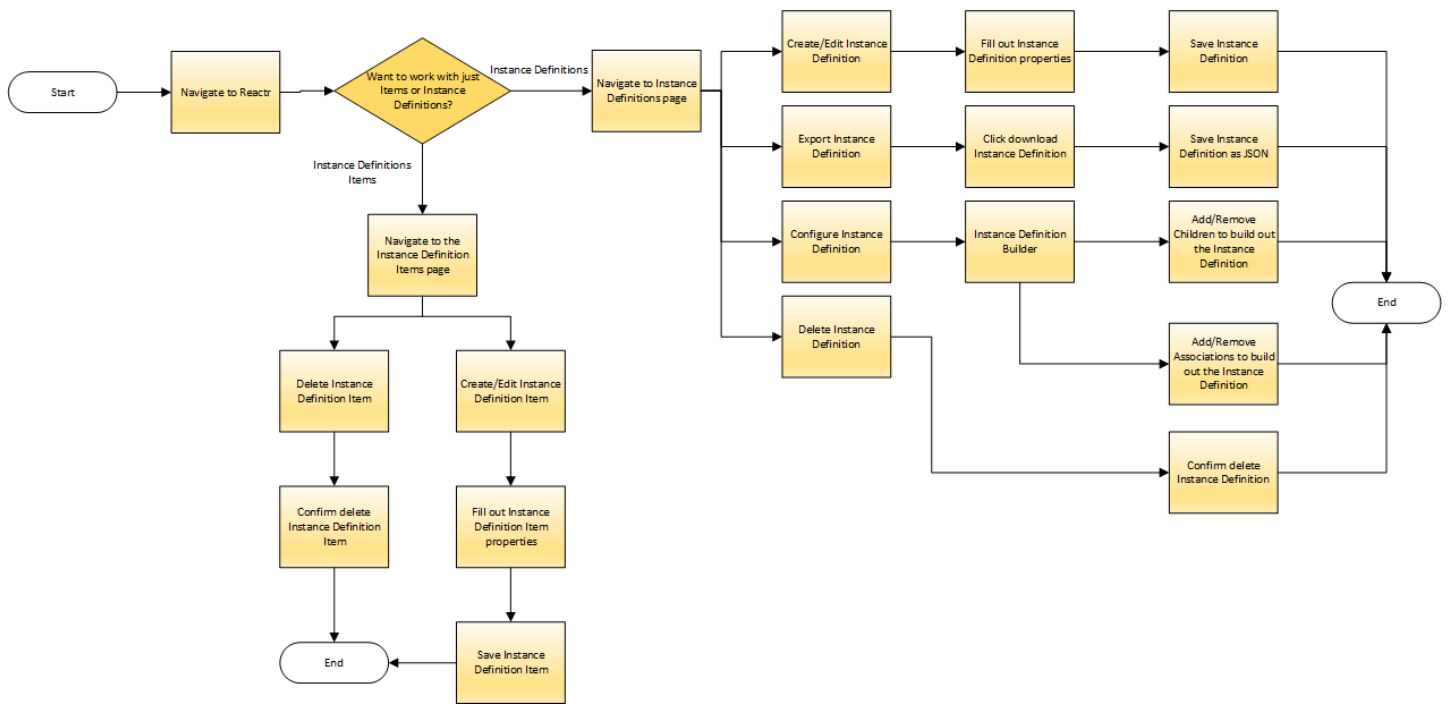
Solution for Generating Instance Definitions

Overview

To simplify the provisioning process instance definitions can be used as templates to reduce the amount of data the user would need to input and reduce the potential for errors.

The solution for creating instance definitions is a new feature as a part of REACTR that enables developers to create instance definitions, create instance definition items and export instance definitions. This new feature is implemented as a web application with a back end that uses Microsoft Azure.

System Workflow

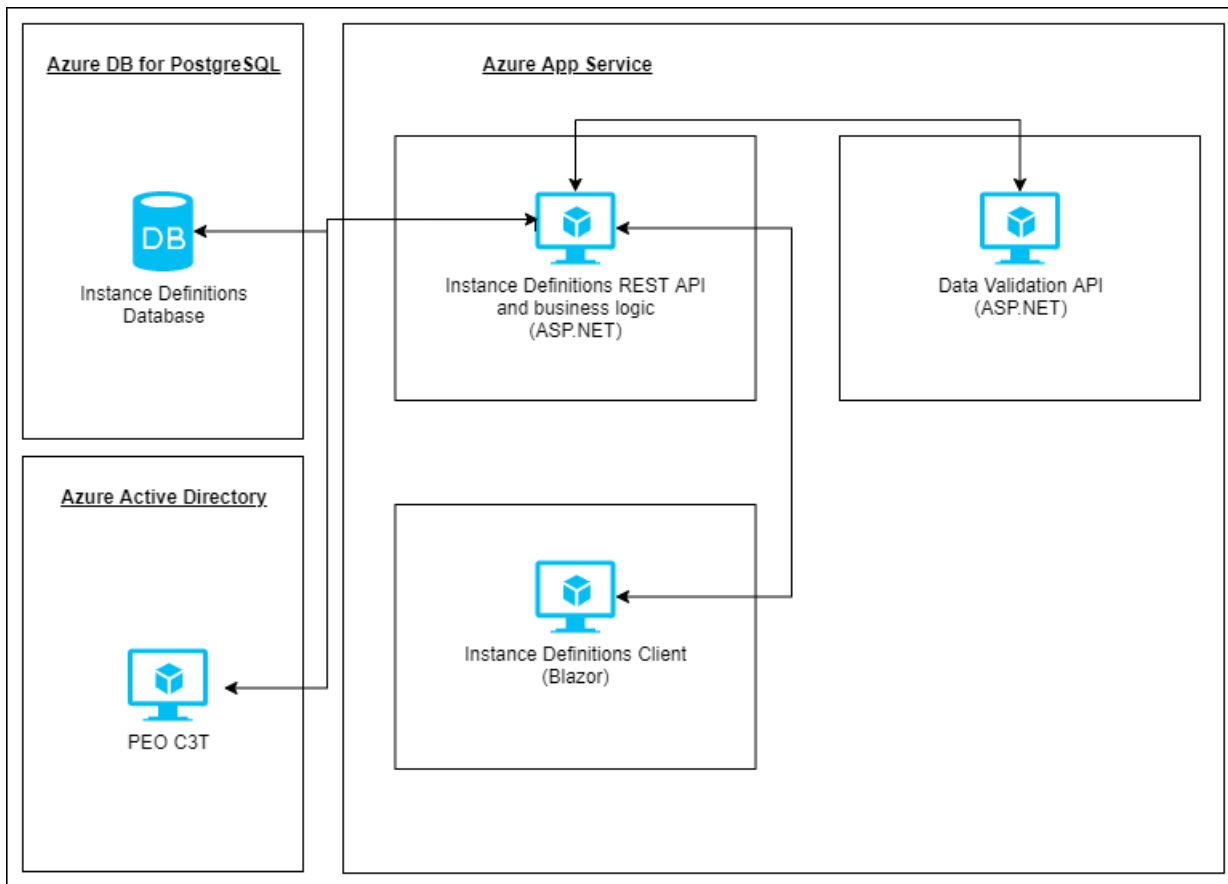


Technologies

TECHNOLOGY NAME	DESCRIPTION	PURPOSE
.NET Core	Open source multi-platform software framework.	Used as the primary base software framework REACTR and its features.
ASP.NET Core	Open source modular web framework.	Used as the web stack for the REACTR web application.
Azure Database for PostgreSQL	IaaS PostgreSQL database service in Azure.	Used as the primary data persistence service for REACTR.

TECHNOLOGY NAME	DESCRIPTION	PURPOSE
JSON	JavaScript Object Notation file format.	Used as the data format for instance definition export files.
ASP.NET Core MVC	Web app framework that implements the model view controller pattern.	Used as the web application stack for the instance definition Items feature in REACTR.
Blazor WebAssembly	Web app framework that enables creation of web apps using C#.	Used as the web application stack for the instance definitions feature in REACTR.
Entity Framework Core	Object relational mapping framework that supports the development of data-oriented applications.	Used as the data layer framework for the instance definitions feature in REACTR.

Architecture



The architecture for the instance definitions feature will consist of adding a new area to the REACTR application that will house all the views, controllers, business logic, and data access logic.

Modified Software

- Area in REACTR to house the instance definitions feature which consists of the controllers, views, business layer, and data access layer.

New Infrastructure

- Azure Database for PostgreSQL to store instance definition Items and instance definitions.

Security

Overview

GROUP NAME	PURPOSE
DEV Instance Definition Admin	Administrator group for instance definition feature in REACTR development site.
PROD Instance Definition Admin	Administrator group for instance definition feature in REACTR UAT and production sites.
DEV Instance Definition Creator	Creator group for instance definition feature in REACTR development site.
PROD Instance Definition Creator	Creator group for instance definition feature in REACTR UAT and production sites.
DEV Instance Definition Viewer	Viewer group for instance definition feature in REACTR development site.
PROD Instance Definition Viewer	Viewer group for instance definition feature in REACTR UAT and production sites.

More detailed information on what permissions are available for viewing specific data or performing specific actions is listed in the feature-specific sections further down in this document.

Persistent Data Design

Data Technical Solution

Instance definition items and instance definitions will be persisted in PostgreSQL. Separate PostgreSQL tables will exist to house the instance definition items instance definitions, and other supporting data.

Data Dictionary

Associations_ResourceResourceGroup

FIELD NAME	C# DATA TYPE	DESCRIPTION
ResourceGroupReferencesId	UUID	Foreign key to ResourceGroupReferences
ResourceReferencesId	UUID	Foreign Key to ResourceReferences

Associations_ResourceTargetGroup

FIELD NAME	C# DATA TYPE	DESCRIPTION
ResourceReferencesId	UUID	Foreign Key to ResourceReferences
TargetGroupReferencesId	UUID	Foreign Key to TargetGroupReferences

Associations_TargetGroupResourceGroup

FIELD NAME	C# DATA TYPE	DESCRIPTION
TargetGroupReferencesId	UUID	Foreign Key to TargetGroupReferences
ResourceGroupReferencesId	UUID	Foreign key to ResourceGroupReferences

Associations_TargetResource

FIELD NAME	C# DATA TYPE	DESCRIPTION
TargetReferencesId	UUID	Foreign Key to TargetReferences
ResourceReferencesId	UUID	Foreign Key to ResourceReferences

Associations_TargetResourceGroup

FIELD NAME	C# DATA TYPE	DESCRIPTION
TargetReferencesId	UUID	Foreign Key to TargetReferences
ResourceGroupReferencesId	UUID	Foreign key to ResourceGroupReferences

Associations_TargetTargetGroup

FIELD NAME	C# DATA TYPE	DESCRIPTION
TargetReferencesId	UUID	Foreign Key to TargetReferences
TargetGroupReferencesId	UUID	Foreign Key to TargetGroupReferences

CachedDataItems

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
Name	CITEXT	Name (or Value depending on how it's used by the application) of the cached data item.
Type	CITEXT	Type of the cached data item.

InstanceDefinitionGrouping

FIELD NAME	C# DATA TYPE	DESCRIPTION
InstanceDefinitionId	UUID	Foreign Key to InstanceDefinitions.
InstanceDefinitionGroupId	UUID	Foreign Key to InstanceDefinitionGroups.

InstanceDefinitionGroups

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
Name	CITEXT	Name of the Instance Definition Group.

InstanceDefinitionItemProperty

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
Name	CITEXT	Name of the Instance Definition Item Property.
Value	CITEXT	Value of the Instance Definition Item Property.
InstanceDefinitionItemId	UUID	Foreign Key to InstanceDefinitionItems.

InstanceDefinitionItemRoles

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
RoleName	CITEXT	Name of the Instance Definition Item Role.
InstanceDefinitionItemId	UUID	Foreign Key to InstanceDefinitionItems.

InstanceDefinitionItems

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
Name	CITEXT	The name of the item from the original item.
EntityName	CITEXT	The value that will become the entity name for the item when the Instance Definition is invoked.
TypeDefinition	CITEXT	The RPS type definition for the item that identifies what the item is and what structure it should have.
IsApproved	BOOLEAN	Boolean that indicates whether the item is approved for use in Instance Definitions. When something IsApproved=\$True, it gets locked and can't be edited unless an administrator unlocks it by setting IsApproved=\$False
IsActive	BOOLEAN	Boolean that indicates whether the item should be active in RPS. IsActive is built into RPS, where IsActive=\$True shows that item as active in RPS. Typically in RPS, when IsActive=\$False, items are ignored - for example if a maintenance window is not active then it will not be used
LastUpdatedBy	CITEXT	The name of the user who updated the item.
LastUpdatedOn	TIMESTAMP	The date/time when the item was updated.
CMDBIDs	TEXT	List of strings that provides a list of RPS CMDB IDs that the item is associated with.
Discriminator	CITEXT	The type name of the item (Resource, Target, ResourceGroup, TargetGroup) that is used as a discriminator value by Entity Framework.
InstanceDefinitionResourceGroup_IsGlobal	BOOLEAN	Boolean that indicates whether the item resource group is to be global in RPS.

InstanceDefinitionNodeProperty

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
Name	CITEXT	Name of the Instance Definition Node Property.
Value	CITEXT	Value of the Instance Definition Node Property.
InstanceDefinitionId	UUID	Foreign Key to InstanceDefinitions.

InstanceDefinitionResourceGroupReference

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
InstanceDefinitionItemId	UUID	Foreign Key to InstanceDefinitionItems.
InstanceDefinitionId	UUID	Foreign Key to InstanceDefinitions.
ParentReferenceId	UUID	Foreign Key that references another record in this table.

InstanceDefinitionResourceReference

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
InstanceDefinitionItemId	UUID	Foreign Key to InstanceDefinitionItems.
InstanceDefinitionId	UUID	Foreign Key to InstanceDefinitions.
ParentReferenceId	UUID	Foreign Key that references another record in this table.

InstanceDefinitionTargetGroupReference

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
InstanceDefinitionItemId	UUID	Foreign Key to InstanceDefinitionItems.
InstanceDefinitionId	UUID	Foreign Key to InstanceDefinitions.
ParentReferenceId	UUID	Foreign Key that references another record in this table.

InstanceDefinitionTargetReference

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
InstanceDefinitionItemId	UUID	Foreign Key to InstanceDefinitionItems.
InstanceDefinitionId	UUID	Foreign Key to InstanceDefinitions.
ParentReferenceId	UUID	Foreign Key that references another record in this table.

InstanceDefinitions

FIELD NAME	C# DATA TYPE	DESCRIPTION
Id	UUID	Primary Key and unique identifier of each record in the table.
Name	CITEXT	Name of the Instance Definition.

FIELD NAME	C# DATA TYPE	DESCRIPTION
IsApproved	BOOLEAN	Boolean that indicates whether the item is approved for use.
Description	CITEXT	Description of the Instance Definition.
RequiredProperties	TEXT	List of required properties for the Instance Definition.
NodeIpAddress	CITEXT	IP Address of the Node.
NodeSyncEndpointUrl	CITEXT	Sync Endpoint URL for the Node.
NodeHostName	CITEXT	Hostname of the Node.
NodeDescription	CITEXT	Description of the Node.
Name	CITEXT	The name of the node from the original node.
EntityName	CITEXT	The value that will become the entity name for the node when the Instance Definition is invoked.
LastUpdatedBy	CITEXT	The name of the user who updated the Instance Definition.
LastUpdatedOn	TIMESTAMP	The date/time when the Instance Definition was updated.

Glossary

TERM	DEFINITION
RPS	Rapid Provisioning System
REACTR	RPS Enterprise Automation, Configuration and Testing Routines
Instance Definition	The abstraction layer on top of existing RPS Types.
Instance Definition Reference	The assignment of the instance definition to a root Resource Item, which results in a set of one or more Resource Items to be run.
Instance Definition Association	The association between referenced items, which results in an assignment between any combination of Resource Items, Target Items, Resource Groups, or Target Groups.
Instance Definition Item	An item on the abstraction layer on top of existing RPS Types.
Instance Definition Node	A node item that will result in association of a node with target items that are created when an instance definition is invoked.
Instance Definition Group	A collection of two or more instance definitions. Groups provide the ability to export multiple instance definitions to a single JSON file.

External Assets

- [Instance Definitions workflow diagrams - PDF](#)
- [Instance Definitions architecture diagram - PDF](#)

Instance Definitions User Guide

Last updated on September 16, 2021.

Document Status: Document Feature Complete as of September 16, 2021; PENDING EXTERNAL REVIEW.

Table of Contents

- Instance Definitions User Guide
 - Table of Contents
 - Introduction
 - Definition of Instance Definitions
 - Purpose of Generating Instance Definitions
 - Solution for Generating Instance Definitions
 - Overview
 - Create or Edit Instance Definition Items
 - Create Instance Definition Items
 - Operations
 - Permissions
 - Edit Instance Definition Items
 - Operations
 - Permissions
 - Approve an Instance Definition Item
 - Operations
 - Permissions
 - Delete an Instance Definition Item
 - Operations
 - Permissions
 - Create or Edit Instance Definitions
 - List Instance Definitions
 - Operations
 - Permissions
 - Create an Instance Definition
 - Operations
 - Permissions
 - Add Tokens to a Property
 - Operations
 - Edit an Instance Definition
 - Operations
 - Permissions
 - Approve or Disapprove an Instance Definition
 - Operations
 - Permissions
 - Copy Instance Definitions
 - Operations
 - Permissions

- Add or Remove Items in an Instance Definition
 - Add an Item to an Instance Definition
 - Operations
 - Permissions
 - Remove an Item from an Instance Definition
 - Operations
 - Permissions
- Validate an Instance Definition or Reference
 - Validation Indicators
 - Validation Results
- Replace Items in an Instance Definition
 - Operations
 - Permissions
- Add or Remove Items From a Referenced Item
 - Add child items
 - Add associations to the item you are currently configuring
 - Operations
 - Permissions
- Delete an Instance Definition
 - Operations
 - Permissions
- Export an Instance Definition
 - Operations
 - Permissions
- Manage Instance Definition Groups
 - Create
 - Delete
 - Edit
 - Export/Download
 - Operations
 - Permissions
- How to use the Token Search
 - Tokens
 - Tokens built into RPS
 - Token Search
 - Bring up the Token Search
 - Use the Token Search
 - Copy Token
 - Append Token
 - Delete Token
- Bulk Replace Tokens
 - Operations
 - Permissions
- How to use Special Characters
 - Supported Special Characters

- [How to Add a Special Character to a Property Value](#)
- [Scenario: How to Create and Export an Instance Definition and its Items](#)
- [Annotated Screenshot Guides](#)
 - [Instance Definitions Listing Annotated Guide](#)
 - [Create or Edit Instance Definition Annotated Guide](#)
 - [Instance Definition Groups Annotated Guide](#)
 - [Create or Edit Instance Definition Group Annotated Guide](#)
 - [Instance Definition Builder Annotated Guide](#)
 - [Add Reference to Instance Definition Annotated Guide](#)
 - [Add Association to Instance Definition Annotated Guide](#)
 - [Change a Referenced Item in Instance Definition Annotated Guide](#)
 - [Instance Definition Items Annotated Guide](#)
 - [Token Picker Annotated Guide](#)
- [Glossary](#)

Introduction

This document describes REACTR's instance definitions feature and provides a user guide. The feature was established based on the need for a way to simplify and improve the Rapid Provision Service's (RPS) provisioning process.

Definition of Instance Definitions

Instance definitions are an abstraction layer on top of existing RPS types. Types currently allow developers define a specific type of resource or target item, its expected or possible Properties and their associated value types (string/int/etc.).

Instance definitions allows developers to take these generic types and combine them together to create complex, nested entities composed of many different types through parent/child and assignment relationships, and the known default values for each individual Types' properties.

Examples: SNEs, TCNs, TSI-Large, laptop, could be a VM (because it could be a collection of components).

There are many existing types such as Vehicle, Computer, VirtualMachine, NetworkConfiguration, etc. The entities soldiers and developers are now interacting with are complex and composed of many of these types in parent/child hierarchies.

Instance definitions are collections of type definitions that have the ability to create an object.

Type definitions are parts with detailed properties. The type definitions do not create objects, but they are able to define something you can create which is what an object will consist of.

Example: Virtual Machine, NIC, drive, etc.

Purpose of Generating Instance Definitions

The RPS provisioning process for a soldier or LSI is limited to selection of a template (instance definition) to use for provisioning and providing unit specific data files, and RPS can successfully provision with this information. The goal is to limit the number of choices and decisions that a soldier or LSI needs to make during provisioning.

By simplifying the RPS provisioning process and removing the number of choices a soldier or LSI needs to make in the field, RPS reduces the likelihood of human error and increases likelihood of adoption due to the improved ease of use.

Solution for Generating Instance Definitions

Overview

To simplify the RPS provisioning process, instance definitions can be used as templates to reduce the amount of data the user would need to input, thereby reducing the potential for errors.

The solution for creating instance definitions is a new feature as a part of REACTR that enables users to create instance definition items, create Instance definition templates, and export those Instance definition templates as JSON files.

This new feature is implemented as a web application with a backend that uses Microsoft Azure.

Create or Edit Instance Definition Items

A user interface to create new instance definition items or edit existing instance definition items.

NOTE

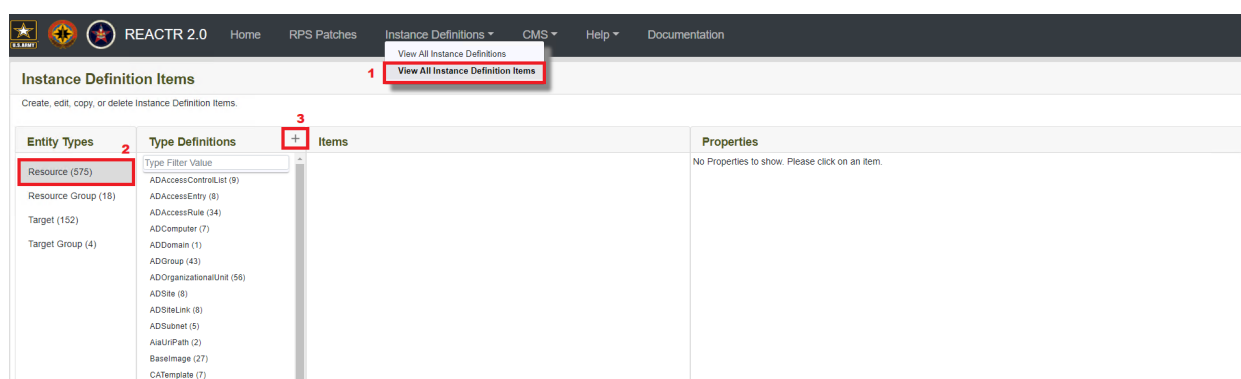
Instance definition items that are 'Approved' cannot be edited.

Once all the necessary instance definition items are created and accurate, they can be used as the toolbox to build instance definitions.

Create Instance Definition Items

The steps necessary to create instance definition items are described below:

1. Navigate to **Instance Definitions > View All Instance Definition Items**
2. Navigate through the list of items by first selecting the **Entity Type**.
3. Select the **Type Definition** that is the type of item you want to add.
 - o If the type definition has a schema available it will be displayed in the list.
 1. Click on the Add (+) button next to the type definition name in the list
 - o If the type definition is not visible then it does not have a schema available. You can add a custom type definition without a schema by following these steps:
 1. Click on the Add (+) button next to the type definitions heading.



2. Type in the name of the type definition.
3. Click the Add (+) button next to the type definition name you entered to add it to the list.



4. Fill out the Properties for the new item.

5. Click on the **Save** button to create the new item.

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

View All Instance Definition Items

Instance Definition Items

Create, edit, copy, or delete Instance Definition Items.

Entity Types 2

- Resource (575)
- Resource Group (18)
- Target (152)
- Target Group (4)

Type Definitions +

Type Filter Value

- ADAccessControlList (9)
- ADAccessEntry (8)
- ADAccessRule (34)
- ADComputer (7)
- ADDomain (1)
- ADGroup (43)
- ADOrganizationalUnit (58)
- ADSite (8)
- ADSiteLink (8)
- ADSubnet (5)
- AltUriPath (2)
- BaseImage (27)
- CATemplate (7)
- CdpUriPath (3)
- Certificate (30)
- CertificationAuthority (1)
- Credential (100)
- Cri (3)
- DataStore (2)
- DnsForwarders (1)
- DSCPPartial (58)
- FirewallRule (2)
- Global (2)
- Host (3)
- IPConfig (2)
- KernelPort (5)
- NpsClient (3)
- NPSPolicyMap (8)
- NtpSource (2)
- OcsppResponder (3)
- OcsppUriPath (3)
- PortGroup (27)
- Stig (20)
- StigException (4)
- StigOrgSetting (2)

Items 3 +

Properties 4

Field Validation Errors

The following fields must be completed correctly before you can continue.

- 'Definition Item Name' is required.

Definition Item Name

Entity Name

Definition Type
Resource

Item Type
ADComputer

Item State
 Not Approved

Is Active

Item Roles

Item Properties [Add Property +](#)

ComputerName Data Mapped

Value

JoinDomain Data Mapped

Value

MemberOfGroup Data Mapped

Value

Path Data Mapped

Value

Cancel Save

⚠ IMPORTANT

Definition item Name is required.

⚠ WARNING

Definition item Name cannot be a duplicate of an existing instance definition.

Click "Add Property" to add item properties to the instance definition item.

Item Properties

[Add Property +](#)

Name (required)

Name



Data Mapped Property Value

Value



⚠ IMPORTANT

Name is required.

⚠ WARNING

Value is required unless Data Mapped Property Value is checked.

Cannot add multiple properties with the same name.

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Item State	Item State toggle	Approves or disapproves an instance definition item so that it becomes uneditable or editable.	Approving an item prevent future changes to the item.
Is Active	Is Active toggle	Sets the IsActive flag on the item.	The IsActive flag is used by RPS to determine whether the item is active or not in RPS.
Add	Add Button	Create a brand-new instance definition item.	The user needs to choose the instance definition item type, type definition, and add values for all required properties for the type selected.
Add Property	Add Property Button	Add a property to an instance definition item.	Property values can be modified after they have been added.

Permissions

OPERATION	ROLE	DESCRIPTION
Create	InstanceDefinitionCreator	The instance definition creator can create instance definitions.
Create	InstanceDefinitionAdmin	The instance definition admin can create instance definitions.
Item State Toggle	InstanceDefinitionAdmin	The instance definition admin can toggle the item state for instance definitions to indicate whether the item is approved or not approved.

Edit Instance Definition Items

The steps necessary to edit instance definition items are described below:

1. Navigate to **Instance Definitions > View All Instance Definition Items**
2. Navigate through the list of items by first selecting the **Entity Type**.
3. Select the **Type Definition** that is the type of item you want to edit.
4. Hover over the item and click on the **Edit (pencil)** button next to the item.
5. Checkmark the desired Roles for the item.
6. Fill out the Property values and add more Properties as needed for the item.
7. Click on the **Save** button to save the item.

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

View All Instance Definitions

View All Instance Definition Items

Instance Definition Items

Create, edit, copy, or delete Instance Definition Items.

Entity Types 2

- Resource (443)
- Resource Group (18)
- Target (129)
- Target Group (4)

Type Definitions

- AA (1)
- ADAccessControlList (1)
- ADComputer (7)
- ADDomain (1)**
- ADGroup (43)
- ADOrganizationalUnit (56)
- ADSite (5)
- ADSiteLink (5)
- ADSubnet (5)
- AiaUriPath (2)
- BaseImage (21)
- CATemplate (7)
- CdplUriPath (3)
- Certificate (30)
- CertificationAuthority (1)
- Credential (42)
- Crl (3)
- CustomType (1)
- DataStore (2)
- DSCPPartial (58)
- Global (2)
- Host (3)
- IPConfig (2)
- KernelPort (5)
- MySchemaTest (1)
- MyTypeDef (1)
- NpsClient (3)
- NPSPolicyMap (8)
- OcspResponder (3)
- OcspUriPath (3)
- PortGroup (24)

ADDomain

Unit-ADDomain

Unit-ADDomain Properties

Definition Item Name [Unit-ADDomain] ✓

Entity Name [*DomainName]

Definition Type Resource

Item Type ADDomain

Item State **Not Approved**

Is Active

Item Roles

Item Properties [Add Property +](#)

Distinguishedname [*DomainDistinguishedName] Data Mapped ✓

ForestName [*DomainName] Data Mapped ✓

FriendlyName [*UnitName] Data Mapped ✓

Path [*DomainDistinguishedName] Data Mapped ✓

UserPrincipalNameSuffix [*UserPrincipalNameSuffix] Data Mapped ✓

Cancel Save

NOTE

If an Instance Definition Item is approved the pencil edit icon will not be selectable. Approved items can be identified by the green lock next to the item name.

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Item State	Item State toggle	Approves or disapproves an instance definition item so that it becomes uneditable or editable.	Approving an item prevent future changes to the item.
Is Active	Is Active toggle	Sets the IsActive flag on the item.	The IsActive flag is used by RPS to determine whether the item is active or not in RPS.
Edit	Edit Button	Allows the user to edit an instance definition item.	Editing an item modify the item stored in the database. These changes modify the item anywhere it is used, across all instance definitions.
Add Property	Add Property Button	Allows the user to add a property to an instance definition item.	Property values can be modified after they have been added.

Permissions

OPERATION	ROLE	DESCRIPTION
Edit	InstanceDefinitionCreator	The instance definition creator can edit instance definitions.
Edit	InstanceDefinitionAdmin	The instance definition admin can edit instance definitions.

OPERATION	ROLE	DESCRIPTION
Item State Toggle	InstanceDefinitionAdmin	The instance definition admin can toggle the item state for instance definitions to indicate whether the item is approved or not approved.

Approve an Instance Definition Item

The user interface provides instance definition administrators multiple methods of approving items. Approving an instance definition item locks the current state of the item so that it is unable to be edited and prevents the item from being deleted.

Item approval functionality is only given to InstanceDefinitionAdmin roles. Items can be approved individually or in bulk by the type of type definition.

The steps necessary to approve an instance definition item are described below:

1. Navigate to **Instance Definitions > View All Instance Definition Items**
2. Select the **Entity Type** containing the **Type Definition** and item you want to approve / disapprove.
3. Select the type of **Type Definition** containing the item, or list of items, you want to approve / disapprove.
 - o Click on the appropriate approval toggle based on the scenario below.
 - Scenario A - Bulk Approval: Select the bulk approval toggle next to the item type header (see screenshot below)
 - This approves all items listed.
 - Scenario B - Individual Item Approval
 - Select the individual item approval toggle next to the item you want to approve.
 - Or, click on the item to view / edit the item for accuracy. Select the individual item approval in the item detail pane.

Instance Definition Item Approval

The screenshot displays the 'Instance Definition Items' interface. On the left, there are panels for 'Entity Types' and 'Type Definitions'. The main area shows a list of 'ADSite' items. A 'Bulk Approval Toggle' is located at the top right of the list, and an 'Individual Item Approval Toggle' is located next to the selected 'BCP-ADSite' item. A 'Current Approval Status' indicator is shown below the selected item. The right-hand pane shows the 'BCP-ADSite Properties' for the selected item, including 'Definition Item Name', 'Last Modified', 'Entity Name', 'Definition Type', 'Item Type', 'Item State', and 'Is Active'.

NOTE

These steps are the same when approving an item from the instance definition builder page or the instance definition item page.

WARNING

When approving an item from the instance definition builder view, you are approving the underlying instance definition item being referenced by the instance definition, not just the reference you created. This means the item will be approved for all instance definitions referencing this specific item.

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Bulk Item State	Bulk Approval Toggle	Bulk approves or disapproves a list of instance definition items by type.	Bulk approving a list of items will prevent future changes to the items.
Item State	Individual Item Approval Toggle	Approves or disapproves an instance definition item so that it becomes uneditable or editable.	Approving an item will prevent future changes to the item.

Permissions

OPERATION	ROLE	DESCRIPTION
Edit	InstanceDefinitionCreator	View current approval status
Edit	InstanceDefinitionAdmin	The instance definition admin has the ability to approve or disapprove instance definition items during edit.
Bulk Approval	InstanceDefinitionAdmin	The instance definition admin can view toggle and bulk approve or unapprove items by type
Item Approval	InstanceDefinitionAdmin	The instance definition admin can view toggle and approve or disapprove items using the item toggle in the type listing

NOTE

Only the instance definition administrators will see the approval toggles. All other roles will still be able to see the status icons showing the current approval status of each item.

Delete an Instance Definition Item

The steps necessary to delete instance definition items are described below:

1. Navigate to **Instance Definitions > View All Instance Definition Items**
2. Navigate through the list of items by first selecting the **Entity Type**.
3. Select the **Type Definition** that is the type of item you want to delete.
4. Click on the **Delete (trashcan)** button.
5. Click on **Delete** button to confirm deletion of the instance definition item.

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

View All Instance Definitions
View All Instance Definition Items

Instance Definition Items

Create, edit, copy, or delete Instance Definition Items.

Entity Types	Type Definitions	ADComputer	Properties
Resource (443) Resource Group (18) Target (129) Target Group (4)	AA (1) ADAccessControlList (1) ADComputer (7) ADDomain (1) ADGroup (43)	BCP-NDM-ADComputer IA-INM-ADComputer JNN-NDM-ADComputer NOP-DCA-ADComputer RVP-ADComputer	No Properties to show. Please click on an item.

Delete Confirmation ✕

Delete?
This will not be recoverable once deleted.

Are you sure you want to delete BCP-NDM-ADComputer?

Cancel Delete

5

NOTE

An instance definition item can only be deleted if it does not have any references or associations to other instance definition items.

CAUTION

A deleted instance definition item is removed permanently.

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Delete	Delete Button	Removes an item from the available instance definition items used for building instance definitions.	Deleting an item removes it permanently from the database.

Permissions

OPERATION	ROLE	DESCRIPTION
Delete	InstanceDefinitionCreator	The instance definition creator can delete instance definitions.
Delete	InstanceDefinitionAdmin	The instance definition admin can delete instance definitions.

Create or Edit Instance Definitions

A user interface to create new instance definitions or edit existing instance definitions.

NOTE

Instance definitions that are 'Approved' cannot be edited.

Once all the necessary references and associations are added to the instance definition, it can be exported and used in RPS.

Validation functionality is in place to validate an entire instance definition's references.

List Instance Definitions

The step necessary to list instance definitions is described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

Instance Definitions View All Instance Definitions
View All Instance Definition Items

Search, filter, sort, create, edit, and delete Instance Definitions.

Instance Definitions Instance Definition Groups

Name	Node Name	Last Modified By	Last Modified Date	Status	Actions
BillID - Simple Instance Definition	MyNode	William DiStefano	7/21/2021 4:41 PM	Not Approved	
BillGlassCI	BillHumVee	Bill Glass	6/16/2021 1:01 AM	Not Approved	
Bravo Team - Test Instance Definition		William DiStefano	6/15/2021 1:55 PM	Not Approved	
Gecho Team - Test NOP-Design7-NIPR-Instance-Definition	NOP-Node	ReactrSystem	6/15/2021 1:43 PM	Not Approved	
JNN-NOP-InstanceDefinition	JNN-NOP-Node	DevInstDefAdmin	6/30/2021 3:36 PM	Not Approved	
NOP-Design7-SIPR-InstanceDefinition	NOP-Node	Paul Chicoine	6/16/2021 2:49 PM	Not Approved	
TCN Lot 7 SIPR	NOP	Barbara Packer	6/15/2021 6:50 PM	Approved	
Test Instance Definition		Devon Aleshire	7/13/2021 11:02 PM	Not Approved	

First Prev 1 Next Last 10 items per page 1 - 8 of 8 items

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Delete	Delete Button	Removes an instance definition.	Deleting an instance definition removes it permanently from the database.
Add	Add Button	Create a brand-new instance definition.	The user needs to fill out the instance definition details and configure it with references to items and associations between items before exporting it to a file.
Edit	Edit Button	Edit an instance definition.	Editing an instance definition modifies the record stored in the database.
Configure	Configure Button	Configure an instance definition.	Configuring an instance definition updates the instance definition items defining the current instance definition configuration
Duplicate	Duplicate button	Duplicate an instance definition.	Duplicating an instance definition duplicates the instance definition and all its references and associations. It does not duplicate the individual instance definition items.
Export	Export button	Export an instance definition.	Exporting an instance definition triggers a download of the exported instance definition to a JSON file.
Status	Status toggle	Approves or disapproves an instance definition.	An approved instance definition is locked and not configurable.

Permissions

OPERATION	ROLE	DESCRIPTION
Add	InstanceDefinitionCreator	The instance definition creator can add instance definitions.
Add	InstanceDefinitionAdmin	The instance definition admin can add instance definitions.
Edit	InstanceDefinitionCreator	The instance definition creator can edit instance definitions.
Edit	InstanceDefinitionAdmin	The instance definition admin can edit instance definitions.
Delete	InstanceDefinitionCreator	The instance definition creator can delete instance definitions.

OPERATION	ROLE	DESCRIPTION
Delete	InstanceDefinitionAdmin	The instance definition admin can delete instance definitions.
Duplicate	InstanceDefinitionCreator	The instance definition creator can duplicate instance definitions.
Duplicate	InstanceDefinitionAdmin	The instance definition admin can duplicate instance definitions.
Export	InstanceDefinitionViewer	The instance definition viewer can export instance definitions.
Export	InstanceDefinitionCreator	The instance definition creator can export instance definitions.
Export	InstanceDefinitionAdmin	The instance definition admin can export instance definitions.
Status	InstanceDefinitionAdmin	The instance definition admin can change the approved status of instance definitions.

Create an Instance Definition

The steps necessary to create an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Create (+)** button on **Actions** column to create a new instance definition.
3. Fill out the details for the instance definition.
4. Click on the **Save** button to save the instance definition.

The screenshot shows the REACTR 2.0 interface. The top navigation bar includes 'Home', 'RPS Patches', 'Instance Definitions', 'CMS', 'Help', and 'Documentation'. The 'Instance Definitions' dropdown menu is open, showing 'View All Instance Definitions' (highlighted with a red box and a '1' annotation) and 'View All Instance Definition Items'. Below the navigation bar, the 'Instance Definitions' page title is displayed. A search bar is present with the text 'Search, filter, sort, create, edit, and delete Instance Definitions.' Below the search bar, the 'Instance Definitions' link is highlighted with a red box and a '2' annotation. The main content area is a table with the following columns: Name, Node Name, Last Modified By, Last Modified Date, Status, and Actions. The 'Actions' column contains a '+' button (highlighted with a red box and a '3' annotation) and several icons for editing and deleting. The table lists 8 instance definitions, each with a status indicator (e.g., 'Not Approved' or 'Approved') and a toggle switch. At the bottom of the table, there is a pagination control showing '10 items per page' and '1 - 8 of 8 items'.

Create Instance Definition ✕

Create Instance Definition
Add/Change Instance Definition detail and properties.

Name (required)
Name ⓘ

Description
Description

Node Fields

Name Node Name	Hostname Host Name
Entity Name Entity Name	IP Address IP Address ✔

Node Description
Description

Sync Endpoint URL
Sync Endpoint URL ✔

Node Properties Add Property +

Close
Save

⚠ IMPORTANT

Name is required and cannot be a duplicate of an existing instance definition name

⚠ IMPORTANT

IP Address and Sync Endpoint URL must have a valid format

⚠ IMPORTANT

If you want an endpoint node to be created then the Name, Entity Name, and IP Address are required, otherwise leave all fields and properties blank.

Click "Add Property" to add node properties to the instance definition.

Node Properties Add Property +

Name (required) ⓘ

Name ⓘ

Data Mapped Property Value

Value ⓘ

⚠ IMPORTANT

Name is required

⚠ WARNING

Value is required unless Data Mapped Property Value is checked

Cannot add multiple properties with the same name

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Save	Save Button	Save the instance definition.	Saves all changes made to the instance definition to the database.
Add Property	Add Property Button	Add a property to the instance definition.	Property values can be modified after they have been added.

Permissions

OPERATION	ROLE	DESCRIPTION
Save	InstanceDefinitionCreator	The instance definition creator can save instance definitions.
Save	InstanceDefinitionAdmin	The instance definition admin can save instance definitions.
Add Property	InstanceDefinitionCreator	The instance definition creator can save instance definitions.
Add Property	InstanceDefinitionAdmin	The instance definition admin can save instance definitions.

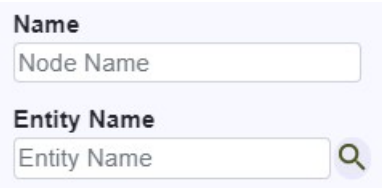
Add Tokens to a Property

The steps necessary to add tokens to a property are described below:


1. Follow steps in [Creating an Instance Definition](#) to get UI for **Create Instance Definition**.
2. Hover over or click inside a text field that can receive a token.

i NOTE

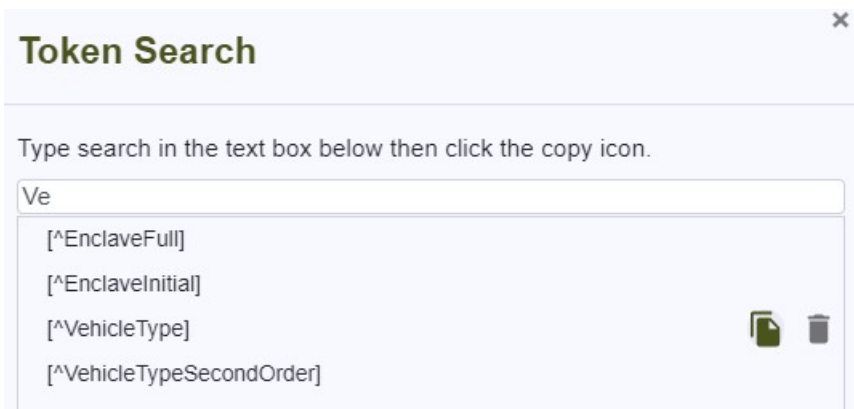
Not all fields are allowed to contain a token.



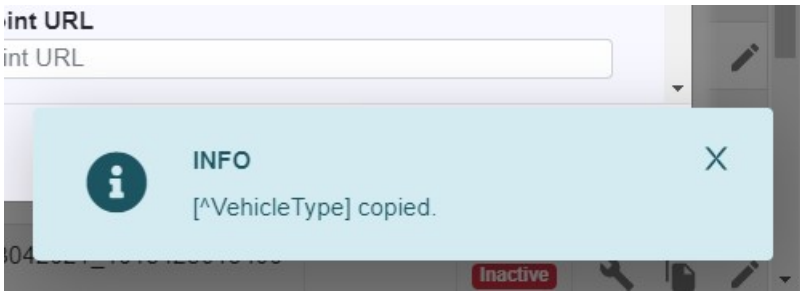
Name

Entity Name
 

3. Click the search icon that appears while the text field has focus.
4. In the popup, a list of available tokens appears. This list can be searched.
5. Hover over the desired choice to see the copy icon.



6. Click the copy icon to copy the token value to the clipboard.



7. Paste the token into the text field using CTRL-V or right-click / paste.

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Copy	Copy Button	Copies the token to the clipboard.	Tokens that are copied to the clipboard can then be pasted into a text field where the user wants the token.
Delete	Delete Button	Deletes the token from the cache.	Deleting a token will only delete it from the cache, it will not remove the token from item fields where it is used.

Edit an Instance Definition

The steps necessary to edit an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Edit (pencil)** button next to the instance definition you want to edit.
3. Fill out the details for the instance definition.
4. Click on the **Save** button to save the instance definition.

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

View All Instance Definitions
View All Instance Definition Items

Instance Definitions

Search, filter, sort, create, edit, and delete Instance Definitions.

Instance Definitions Instance Definition Groups

Name	Node Name	Last Modified By	Last Modified Date	Status	Actions
Search name	Search node name				
BLID - Simple Instance Definition	MyNode	William DiStefano	7/21/2021 4:41 PM	Not Approved	
BillGlassCI	BillHumVee	Bill Glass	6/16/2021 1:01 AM	Not Approved	
Bravo Team - Test Instance Definition		William DiStefano	6/15/2021 1:55 PM	Not Approved	
Gecho Team - Test NOP-Design7-NIPR-Instance-Definition	NOP-Node	ReactrSystem	6/15/2021 1:43 PM	Not Approved	
JNN-NOP-InstanceDefinition	JNN-NOP-Node	DevInstDefAdmin	6/30/2021 3:36 PM	Not Approved	
NOP-Design7-SIPR-InstanceDefinition	NOP-Node	Paul Chicoine	6/16/2021 2:49 PM	Not Approved	
TCN Lot 7 SIPR	NOP	Barbara Packer	6/15/2021 6:50 PM	Approved	
Test Instance Definition		Devon Aleshire	7/13/2021 11:02 PM	Not Approved	

First Prev 1 Next Last 10 items per page 1 - 8 of 8 items

Edit Instance Definition

Edit Instance Definition
Add/Change Instance Definition detail and properties.

Name (required)
Bravo Team - Test Instance Definition ✓

Description
Description

Node Fields

Name
Node Name

Hostname
Host Name

Entity Name
Entity Name

IP Address
IP Address ✓

Node Description
Description

Sync Endpoint URL
Sync Endpoint URL ✓

Node Properties [Add Property +](#)

test Data Mapped

test ✓

[Close](#) [Save](#)

When an error occurs, a summary of page errors will appear. The fields will also contain visual indicators.

Edit Instance Definition

Field Validation Errors
The following fields must be completed correctly before you can continue.

- Name is required.

Name (required)
Name

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Save	Save Button	Save the instance definition.	Saves all changes made to the instance definition to the database.
Add Property	Add Property Button	Add a property to the instance definition.	Property values can be modified after they have been added.

Permissions

OPERATION	ROLE	DESCRIPTION
Save	InstanceDefinitionCreator	The instance definition creator can save instance definitions.
Save	InstanceDefinitionAdmin	The instance definition admin can save instance definitions.
Add Property	InstanceDefinitionCreator	The instance definition creator can save instance definitions.
Add Property	InstanceDefinitionAdmin	The instance definition admin can save instance definitions.

Approve or Disapprove an Instance Definition

NOTE

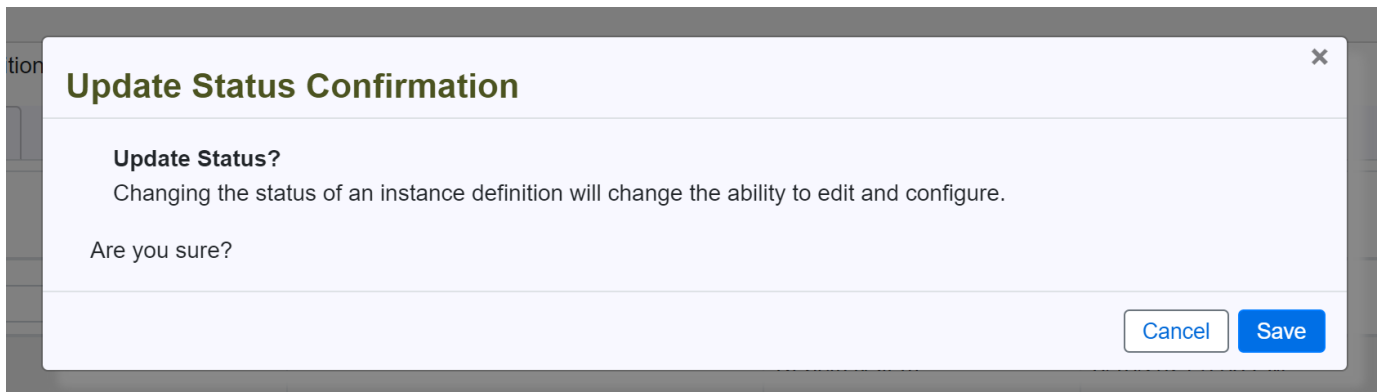
When an instance definition is approved, it cannot be edited, deleted, or configured.

The steps necessary to approve or disapprove an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Status toggle** for the instance definition that you want to update.
3. Click on the **Save** button on the confirmation dialog to save the instance definition status.

The screenshot shows the REACTR 2.0 web interface. The top navigation bar includes 'Instance Definitions' and 'View All Instance Definitions'. Below the navigation, there is a search bar and a table of instance definitions. The table has columns: Name, Node Name, Last Modified By, Last Modified Date, Status, and Actions. The 'Bravo Team - Test Instance Definition' row is selected. The 'Status' column for this row shows 'Not Approved' with a toggle switch. A red box highlights this toggle. A confirmation dialog is open over the toggle, with a red box highlighting the 'Save' button. The page footer shows '1 - 8 of 8 items'.

When the toggle is updated, a confirmation dialog will appear asking for confirmation of the update.



Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Status	Status toggle	Approves or disapproves the instance definition so that it becomes uneditable or editable.	Approving an instance definition will prevent future changes to the instance definition.
Save	Save Button	Save the instance definition's status.	Saves the changes made to the instance definition to the database.

Permissions

OPERATION	ROLE	DESCRIPTION
Approve/disapprove	InstanceDefinitionAdmin	The instance definition admin can approve or disapprove instance definitions.

Copy Instance Definitions

A user interface to copy instance definitions.

NOTE

Copying an instance definition makes an exact copy of the instance definition and all its references and associations.

Copying an instance definition does not copy the individual instance definition items.

The steps necessary to copy an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Copy (paper)** button next to the instance definition you want to copy.

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

View All Instance Definitions
View All Instance Definition Items

Instance Definitions

Search, filter, sort, create, edit, and delete Instance Definitions.

Instance Definitions Instance Definition Groups

Name	Node Name	Last Modified By	Last Modified Date	Status	Actions
Search name	Search node name				
BIID - Simple Instance Definition	MyNode	William DiStefano	7/21/2021 4:41 PM	Not Approved	
BillGlassCI	BillHumVee	Bill Glass	6/16/2021 1:01 AM	Not Approved	
Bravo Team - Test Instance Definition		William DiStefano	6/15/2021 1:55 PM	Not Approved	
Gecho Team - Test NOP-Design7-NIPR-Instance-Definition	NOP-Node	ReactrSystem	6/15/2021 1:43 PM	Not Approved	
JNN-NOP-InstanceDefinition	JNN-NOP-Node	DevInstDefAdmin	6/30/2021 3:36 PM	Not Approved	
NOP-Design7-SIPR-InstanceDefinition	NOP-Node	Paul Chicoine	6/16/2021 2:49 PM	Not Approved	
TCN Lot 7 SIPR	NOP	Barbara Packer	6/15/2021 6:50 PM	Approved	
Test Instance Definition		Devon Aleshire	7/13/2021 11:02 PM	Not Approved	

First Prev 1 Next Last 10 items per page 1 - 8 of 8 items

3. Enter the Copy Name for the instance definition you are copying and click on the **Copy** button.

Copy Instance Definition

Copy
Duplicate this Instance Definition with a new name.

Field Validation Errors
The following fields must be completed correctly before you can continue.

- Copy Name is required.

Copy Name (required)
Name of copy

Cancel Copy

IMPORTANT

Copy Name is required.

WARNING

Copy Name cannot be a duplicate of an existing instance definition that has the same Name.

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Copy	Copy button	Copy an instance definition.	Copying an instance definition copies the instance definition and all its references and associations. This does not copy the individual instance definition items.

Permissions

OPERATION	ROLE	DESCRIPTION

OPERATION	ROLE	DESCRIPTION
Copy	InstanceDefinitionCreator	The instance definition creator can copy instance definitions.
Copy	InstanceDefinitionAdmin	The instance definition admin can copy instance definitions.

Add or Remove Items in an Instance Definition

A user interface to add or remove instance definition items in an instance definition.

When an instance definition item is added to an instance definition it is not directly added; rather, it is referenced in the instance definition or associated to another referenced instance definition item.

When an instance definition item is added:

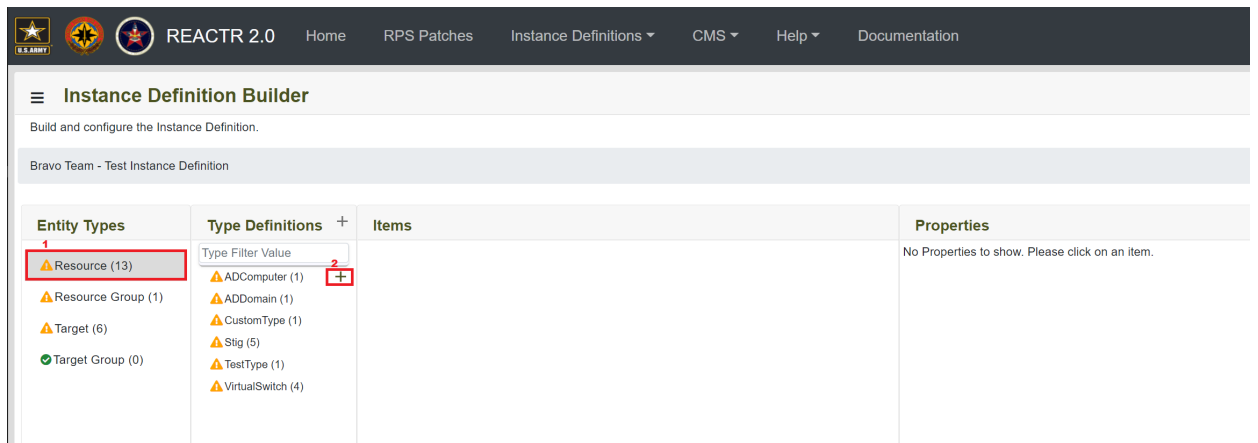
- At the parent/root instance definition level: it is added as a **reference**.
- Under a referenced instance definition item of the **same** entity type: it is added as a **reference**.
- Under a referenced instance definition item of a **different** entity type: it is added as an **association**.

Add an Item to an Instance Definition

The steps necessary to add an item to an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Configure (wrench)** button next to the instance definition you want to configure/build.
3. Navigate through the instance definition by first selecting the **Entity Type**.
4. Select the **Type Definition** that is related to the item you want to add.

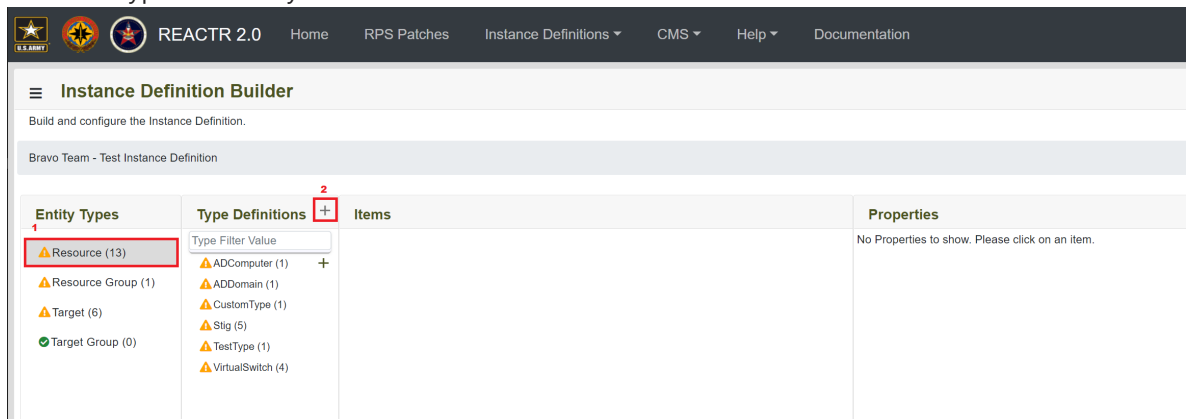
- If the type definition is visible:
 1. Click on the **Add (+)** button next to the type definition.
 2. Choose the items you want to reference in the instance definition.



o If the type definition is not visible:

1. Click on the **Add (+)** button next to the column header **Type Definitions**.

1. Select the type definition you want.



5. Choose the items you want to reference in the instance definition.

6. Click the **Select** button.

Add Reference (BaselImage)

Add References

Add references to another item below. Select one or more items.

Items	ASC-BaselImage Properties
<input checked="" type="checkbox"/> ASC-BaselImage <input type="checkbox"/> Bootstrap_BaselImage <input type="checkbox"/> CMX-BaselImage <input type="checkbox"/> DCA-BaselImage <input type="checkbox"/> DSC-BaselImage <input type="checkbox"/> FWH-BaselImage <input type="checkbox"/> FWM-BaselImage <input type="checkbox"/> HSM-BaselImage <input type="checkbox"/> INM-BaselImage <input type="checkbox"/> NOM-BaselImage <input type="checkbox"/> NRM-BaselImage <input type="checkbox"/> NST-BaselImage <input type="checkbox"/> NTP-BaselImage <input type="checkbox"/> PEP-BaselImage <input type="checkbox"/> RVP-BaselImage <input type="checkbox"/> Server2012R2-BaselImage <input type="checkbox"/> Server2012r2-IsolImage <input type="checkbox"/> Server2019-BaselImage <input type="checkbox"/> Server2019-IsolImage <input type="checkbox"/> SIM1-BaselImage <input type="checkbox"/> T2R-BaselImage	Definition Item Name ASC-BaselImage Last Modified Matthew Collera - 3/19/2021 5:20 PM Entity Name ASC-BaselImage Definition Type Resource Item Type BaselImage Item State Not Approved <input type="checkbox"/> Is Active Active Item Roles Item Properties Architecture x64 FileName ASC_5_14_1.ovf ImageName ASC ImageType OVF OsType
<input type="button" value="Select"/> <input type="button" value="Cancel"/>	

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Add	Add button	Add a reference to an instance definition item or association between instance definition items.	If the parent is the root instance definition or a referenced instance definition item of the same entity type, then a reference will be created. Otherwise, an association will be created between the item being added and the parent item.

Permissions

OPERATION	ROLE	DESCRIPTION
Add	InstanceDefinitionCreator	The instance definition creator can add references to instance definition items or associations between instance definition items.
Add	InstanceDefinitionAdmin	The instance definition admin can add references to instance definition items or associations between instance definition items.

Remove an Item from an Instance Definition

The steps necessary to remove an item to an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Configure (wrench)** button next to the instance definition you want to configure/build.

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

Instance Definitions
View All Instance Definitions
View All Instance Definition Items

Search, filter, sort, create, edit, and delete Instance Definitions.

Instance Definitions Instance Definition Groups

Name	Node Name	Last Modified By	Last Modified Date	Status	Actions
BillID - Simple Instance Definition	MyNode	William DiStefano	7/21/2021 4:41 PM	Not Approved	[Icons]
BillGlassCI	BillHumVee	Bill Glass	6/16/2021 1:01 AM	Not Approved	[Icons]
Bravo Team - Test Instance Definition		William DiStefano	6/15/2021 1:55 PM	Not Approved	[Icons]
Gecho Team - Test NOP-Design7-NIPR-Instance-Definition	NOP-Node	ReactrSystem	6/15/2021 1:43 PM	Not Approved	[Icons]
JNN-NOP-InstanceDefinition	JNN-NOP-Node	DevInstDefAdmin	6/30/2021 3:36 PM	Not Approved	[Icons]
NOP-Design7-SIPR-InstanceDefinition	NOP-Node	Paul Chicoine	6/16/2021 2:49 PM	Not Approved	[Icons]
TCN Lot 7 SIPR	NOP	Barbara Packer	6/15/2021 6:50 PM	Approved	[Icons]
Test Instance Definition		Devon Aleshire	7/13/2021 11:02 PM	Not Approved	[Icons]

First Prev 1 Next Last 10 items per page 1 - 8 of 8 items

3. Navigate through the instance definition by first selecting the **Entity Type**.
4. Select the **Type Definition** that is related to the item you want to remove.
5. Click on **Trash** icon to delete.

REACTR 2.0 Home RPS Patches Instance Definitions CMS Help Documentation

Instance Definition Builder
Build and configure the Instance Definition.

Bravo Team - Test Instance Definition

Entity Types

- Resource (13)
- Resource Group (1)
- Target (6)
- Target Group (0)

Type Definitions

Type Filter Value

- ADComputer (1)
- ADDomain (1)
- CustomType (1)
- Stig (5)
- TestType (1)
- VirtualSwitch (4)

ADComputer

- RVP-ADComputer

RVP-ADComputer Properties

Definition Item Name
RVP-ADComputer

Last Modified
ReactrSystem - 9/8/2021 9:13 PM

Entity Name
[^\VehicleType][^\TelephonyId][^\EnclaveInitial]RVP

Definition Type
Resource

Item Type
ADComputer

Item State
Not Approved

Is Active
Active

Item Roles

Item Properties

ComputerName
[^\VehicleType][^\TelephonyId][^\EnclaveInitial]RVP

JoinDomain
True

MemberOfGroup
WIN-T-Online-Responder-Hosts

Path
OU=Computers,OU=RPS,[^\DomainDistinguishedName]

6. Confirm that you want to delete the referenced item from the instance definition.

Delete Confirmation

Delete?
This will not be recoverable once deleted.

Are you sure?

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Remove	Remove button	Remove a reference to an instance definition item or association between instance definition items.	If the parent is the root instance definition or a referenced instance definition item of the same entity type, then a reference will be removed. Otherwise, an association will be removed between the item being removed and the parent item.

Permissions

OPERATION	ROLE	DESCRIPTION
Remove	InstanceDefinitionCreator	The instance definition creator can remove references to instance definition items or associations between instance definition items.
Remove	InstanceDefinitionAdmin	The instance definition admin can remove references to instance definition items or associations between instance definition items.

Validate an Instance Definition or Reference

Validation Overview

Validation of instance definitions works by validating a full referenced item against its type definition schema. A full referenced item consists of:

- referenced item's entity name
- referenced item's properties
- referenced item's roles
- referenced item's children (to other referenced items)
- referenced item's associations (to other referenced items)

Validation is performed by taking the above information and comparing it against the type definition schema of the parent referenced item. If there are any discrepancies then these validation results will be available for you to see by clicking on the validation errors button. Clicking on the validation errors button will open the Validation Results modal.

The screenshot shows the Reactr 2.0 Instance Definition Builder interface. The top navigation bar includes links for Home, RPS Patches, Instance Definitions, CMS, Help, and Documentation. The main content area displays the Instance Definition Builder for 'JNN-NOP-InstanceDefinition'. A red box highlights a message in the top right corner that reads '338 ERRORS (CLICK TO VIEW)'.

Validation Indicators

A status message displays the current validation status of the instance definition or reference. The following indicators are displayed based on the status of the validation:

Not validated - Indicates that the validation process has not started. This process is automatic.

NOT VALIDATED

Validating - Indicates that the validation process started but has not completed. The screen will update as the validation completes.

VALIDATING, PLEASE WAIT

Errors - Indicates that there are 1 or more errors. Click on the message to see the detail validation messages.

122 ERRORS (CLICK TO VIEW)

Valid - Indicates that no validation errors are detected.

REFERENCE IS VALID

Validation Results

Detailed validation messages are show with the validation results. Across the top of the page are tabs based on the type of validation error.

Validation Errors

ValidationError (8)	MissingProperty (109)	DeserializationError (5)
---------------------	-----------------------	--------------------------

- ADComputer data validation error: The ComputerName field is required.
- Computer data validation error: The BaseImage field is required.
- Computer data validation error: The NIC field is required.
- VirtualMachine data validation error: The BaseImage field is required.
- VirtualMachine data validation error: The Credential field is required.
- VirtualMachine data validation error: The Host field is required.
- VirtualMachine data validation error: The NIC field is required.
- VirtualMachine data validation error: The VMId field is required.

Fixing Validation Errors

To fix a validation error you may need to alter any of the following:

1. Missing, invalid, or extraneous properties on the Item.
2. Missing, invalid, or extraneous properties for a role on the Item.
3. Missing, invalid, or extraneous children or associations for the Item.
4. Missing, invalid, or extraneous properties for any children or associations directly below the parent Item.

TIP

If the error "**An error occurred while validating** <Item Type>" occurs, make sure the type schema does not contain a property named *ExtraProperties*.

Replace Items in an Instance Definition

A user interface to replace instance definition items in an instance definition.

IMPORTANT

When an item is replaced for a different item, it must be replaced with an item of the same entity type and type definition.

The steps necessary to replace items in an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**

2. Click on the **Configure (wrench)** button next to the instance definition you want to configure/build.

Instance Definitions

Search, filter, sort, create, edit, and delete Instance Definitions.

Instance Definitions Instance Definition Groups

Name	Node Name	Last Modified By	Last Modified Date	Status	Actions
Search name	Search node name				
BillID - Simple Instance Definition	MyNode	William DiStefano	7/21/2021 4:41 PM	Not Approved	
BillGlassCI	BillHumVee	Bill Glass	6/16/2021 1:01 AM	Not Approved	
Bravo Team - Test Instance Definition		William DiStefano	6/15/2021 1:55 PM	Not Approved	
Gecho Team - Test NOP-Design7-NIPR-Instance-Definition	NOP-Node	ReactrSystem	6/15/2021 1:43 PM	Not Approved	
JNN-NOP-InstanceDefinition	JNN-NOP-Node	DevInstDefAdmin	6/30/2021 3:36 PM	Not Approved	
NOP-Design7-SIPR-InstanceDefinition	NOP-Node	Paul Chicoine	6/16/2021 2:49 PM	Not Approved	
TCN Lot 7 SIPR	NOP	Barbara Packer	6/15/2021 6:50 PM	Approved	
Test Instance Definition		Devon Aleshire	7/13/2021 11:02 PM	Not Approved	

First Prev 1 Next Last 10 items per page 1 - 8 of 8 items

3. Navigate through the instance definition by first selecting the **Entity Type**.

4. Select the **Type Definition** that is related to the item you want to remove.

5. Click on the **Replace (swap arrows)** button next to the referenced item that you want to replace.

Instance Definition Builder

Build and configure the Instance Definition.

Bravo Team - Test Instance Definition

Entity Types	Type Definitions	ADComputer	Properties
Resource (13) Resource Group (1) Target (6) Target Group (0)	Type Filter Value ADComputer (1) ADDomain (1) CustomType (1) Stig (5) TestType (1) VirtualSwitch (4)	RVP-ADComputer 	No Properties to show. Please click on an item.

6. Select (click on) the item that you want to replace it with.

7. Confirm that you want to replace the referenced item in the instance definition by clicking the **Select** button.

Change Reference (ADComputer)

Swap References

Select reference to swap.

Items	1. Click on the item to use as the replacement	BCP-NDM-ADComputer Properties
	<p>BCP-NDM-ADComputer</p> <p>IA-INM-ADComputer</p> <p>JNN-NDM-ADComputer</p> <p>NOP-DCA-ADComputer</p> <p>Testing11</p> <p>Testing322222222</p>	<p>Definition Item Name BCP-NDM-ADComputer</p> <p>Last Modified ReactrSystem - 9/8/2021 9:13 PM</p> <p>Entity Name BCP[^TelephonyId][^EnclaveInitial]NDM</p> <p>Definition Type Resource</p> <p>Item Type ADComputer</p> <p>Item State Not Approved</p> <p>Is Active Active</p> <p>Item Roles</p> <p>Item Properties</p> <p>ComputerName BCP[^TelephonyId][^EnclaveInitial]NDM</p> <p>JoinDomain True</p> <p>Path OU=BCP Platforms,[^DomainDistinguishedName]</p>
	<p>2. Click on the Select Button</p> <p>Select Cancel</p>	

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Replace	Replace button	Replace a reference to an instance definition item with another instance definition item of the same entity type and type definition.	

Permissions

OPERATION	ROLE	DESCRIPTION
Replace	InstanceDefinitionCreator	The instance definition creator can replace references to instance definition items.
Replace	InstanceDefinitionAdmin	The instance definition admin can replace references to instance definition items.

Add or Remove Items From a Referenced Item

A user interface to add child items under a parent item or make associations between items.

When an instance definition item is added to an instance definition it is not directly added; rather, it is referenced in the instance definition, or associated to another referenced instance definition item.

When an instance definition item is added:

- Under a referenced instance definition item of the **same** entity type: it is added as a **reference**.

- Under a referenced instance definition item of a **different** entity type: it is added as an **association**.

The steps necessary to add or remove items from a referenced item are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Configure (wrench)** button next to the instance definition you want to configure/build.
3. Navigate through the instance definition by first selecting the **Entity Type**.
4. Select the **Type Definition** that is related to the item you want to configure/build.
5. Click on the **Configure (wrench)** button next to the referenced item you want to configure/build.

Instance Definitions

Search, filter, sort, create, edit, and delete Instance Definitions.

Instance Definitions Instance Definition Groups

Name	Node Name	Last Modified By	Last Modified Date	Status	Actions
BillID - Simple Instance Definition	MyNode	William DiStefano	7/21/2021 4:41 PM	Not Approved	[Icons]
BillGlassCI	BillHumVee	Bill Glass	6/16/2021 1:01 AM	Not Approved	[Icons]
Bravo Team - Test Instance Definition		William DiStefano	6/15/2021 1:55 PM	Not Approved	[Icons]
Gecho Team - Test NOP-Design7-NIPR-Instance-Definition	NOP-Node	ReactrSystem	6/15/2021 1:43 PM	Not Approved	[Icons]
JNN-NOP-InstanceDefinition	JNN-NOP-Node	DevInstDefAdmin	6/30/2021 3:36 PM	Not Approved	[Icons]
NOP-Design7-SIPR-InstanceDefinition	NOP-Node	Paul Chicoine	6/16/2021 2:49 PM	Not Approved	[Icons]
TCN Lot 7 SIPR	NOP	Barbara Packer	6/15/2021 6:50 PM	Approved	[Icons]
Test Instance Definition		Devon Aleshire	7/13/2021 11:02 PM	Not Approved	[Icons]

10 items per page

1 - 8 of 8 items

Instance Definition Builder

Build and configure the Instance Definition.

Bravo Team - Test Instance Definition

Entity Types	Type Definitions	ADComputer	Properties
<ul style="list-style-type: none"> Resource (13) Resource Group (1) Target (6) Target Group (0) 	Type Filter Value <ul style="list-style-type: none"> ADComputer (1) ADDomain (1) CustomType (1) Stig (5) TestType (1) VirtualSwitch (4) 	RVP-ADComputer [Configure]	No Properties to show. Please click on an item.

Instance Definition Builder

Build and configure the Instance Definition.

Bravo Team - Test Instance Definition / RVP-ADComputer

Entity Types	Type Definitions	ADComputer	Properties
<ul style="list-style-type: none"> Children (0) Resource Group (1) Target (0) Target Group (0) 	Type Filter Value ADComputer		No Properties to show. Please click on an item.

NOTE

When you configure a referenced item, notice that the breadcrumb at the top of the page now displays the referenced item's name. You are now

configuring that specific referenced item's children references and associations.

Add child items

1. Click on the children entity type.
2. Follow these instructions for how to add or remove referenced items: [Adding or Removing Items in an Instance Definition](#)

Add associations to the item you are currently configuring

1. Click on any entity type except for children. The children entity type refers to parent/child referenced item relationships.
2. Follow these instructions for how to add or remove referenced items: [Adding or Removing Items in an Instance Definition](#)

NOTE

Any references added like this will become associations that are associated with the referenced item you are currently configuring.

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Add	Add button	Add a reference to an instance definition item or association between instance definition items.	If the parent is the root instance definition or a referenced instance definition item of the same entity type, then a reference will be created. Otherwise, an association will be created between the item being added and the parent item.
Remove	Remove button	Removes a reference to an instance definition item or association between instance definition items.	If the parent is the root instance definition or a referenced instance definition item of the same entity type, then a reference will be removed. Otherwise, an association will be removed between the item being removed and the parent item.

Permissions

OPERATION	ROLE	DESCRIPTION
Add	InstanceDefinitionCreator	The instance definition creator can add references to instance definition items or associations between instance definition items.
Add	InstanceDefinitionAdmin	The instance definition admin can add references to instance definition items or associations between instance definition items.
Remove	InstanceDefinitionCreator	The instance definition creator can remove references to instance definition items or associations between instance definition items.
Remove	InstanceDefinitionAdmin	The instance definition admin can remove references to instance definition items or associations between instance definition items.

Delete an Instance Definition

A user interface to delete an instance definition.

WARNING

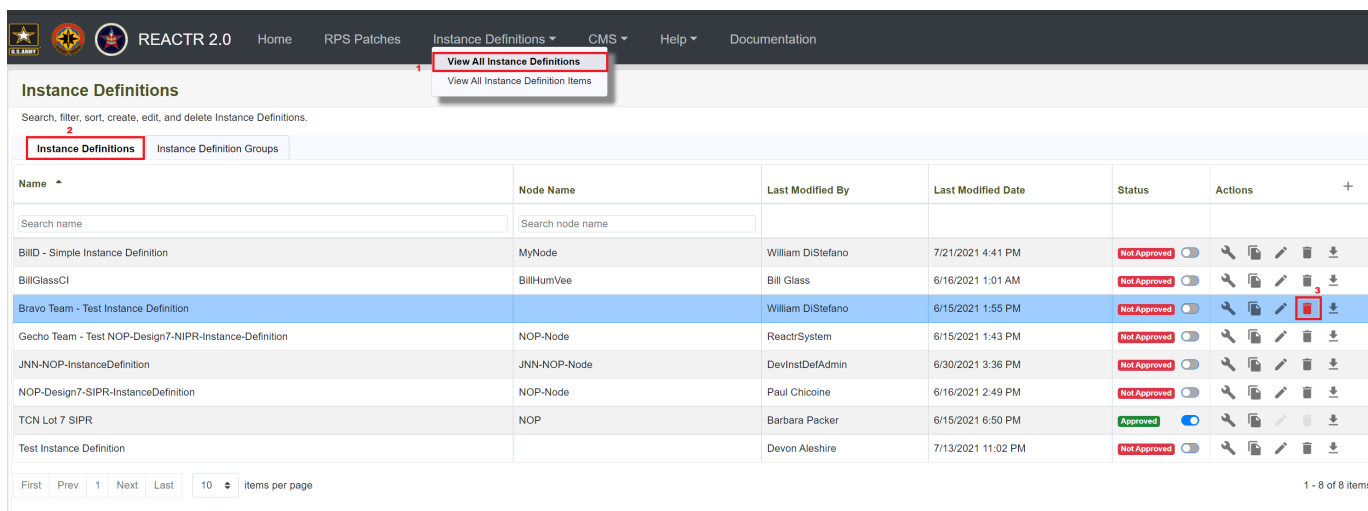
Deleting an instance definition will permanently delete the instance definition, and any references or associations that it contained.

NOTE

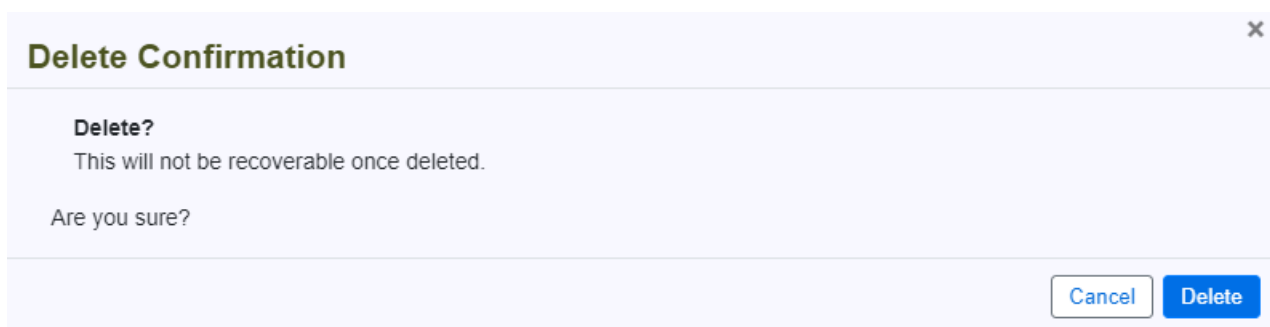
This does not delete the actual instance definition items that were referenced or associated in the deleted instance definition.

The steps necessary to delete an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Delete (trashcan)** button next to the instance definition you want to delete.



3. Confirm that you want to delete the instance definition.



Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Delete	Delete Button	Removes an instance definition.	Deleting an instance definition removes it permanently from the database.

Permissions

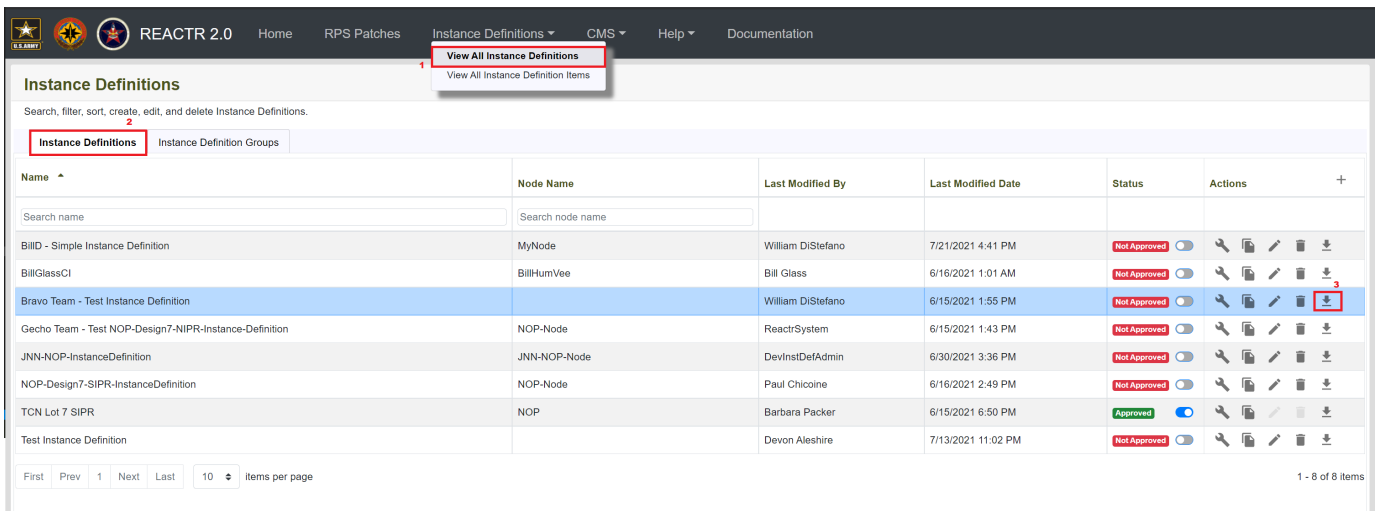
OPERATION	ROLE	DESCRIPTION
Delete	InstanceDefinitionCreator	The instance definition creator can delete instance definitions.
Delete	InstanceDefinitionAdmin	The instance definition admin can delete instance definitions.

Export an Instance Definition

A user interface to export an instance definition to a JSON file. The steps necessary to export an instance definition are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**

2. Click on the **Export (down arrow)** button next to the instance definition you want to export.



Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Export	Export button	Export an instance definition.	Exporting an instance definition will trigger a download of the exported instance definition to a JSON file.

Permissions

OPERATION	ROLE	DESCRIPTION
Export	InstanceDefinitionViewer	The instance definition viewer can export instance definitions.
Export	InstanceDefinitionCreator	The instance definition creator can export instance definitions.
Export	InstanceDefinitionAdmin	The instance definition admin can export instance definitions.

Manage Instance Definition Groups

A user interface to create, edit, delete, and export instance definition groups. An instance definition group is a collection of two or more instance definitions.

The primary purpose of this functionality is to provide the ability to export multiple instance definitions to a single JSON file.

The steps necessary to manage instance definition groups are described below:

1. Navigate to **Instance Definitions > View All Instance Definitions**
2. Click on the **Instance Definition Groups** tab.

Create

1. Click on the **Create (+)** button at the top of the grid to create a new group.
2. Enter a name for the group.
3. Check the instance definitions that you want in the group.
4. Click on the **Save** button to save your changes.

Delete

1. Click on the **Delete (trashcan)** button next to the instance definition group you want to delete.

Edit

1. Click on the **Edit (pencil)** button next to the instance definition group you want to edit.
2. Check the instance definitions that you want in the group.
3. Click on the **Save** button to save your changes.

Export/Download

1. Click on the **Export (down arrow)** button next to the instance definition group you want to export.

Instance Definitions

Search, filter, sort, create, edit, and delete Instance Definitions.

Instance Definitions | **Instance Definition Groups**

Instance Definition Groups | Test Group 0

Group Name ^	Last Modified By	Last Modified Date	Actions +
Search Group			
Test Group 0	ReactrSystem	05/18/2021 12:35:25	
Test Group 1	ReactrSystem	05/18/2021 12:35:25	
Test Group 10	ReactrSystem	05/18/2021 12:35:25	
Test Group 2	ReactrSystem	05/18/2021 12:35:25	
Test Group 3	ReactrSystem	05/18/2021 12:35:25	
Test Group 4	ReactrSystem	05/18/2021 12:35:25	
Test Group 5	ReactrSystem	05/18/2021 12:35:25	
Test Group 6	ReactrSystem	05/18/2021 12:35:25	
Test Group 7	ReactrSystem	05/18/2021 12:35:25	
Test Group 8	ReactrSystem	05/18/2021 12:35:25	

First Prev 1 2 Next Last 10 items per page 1 - 10 of 12 items

Edit Instance Definition Group

Name: Really Awesome Group

Instance Definitions

- JNN-NOP-InstanceDefinition-Copy
- Testing Entity Rework
- 46hhhj778yuhj
- Chris 33
- Chris 33-Copy
- Chris 33-Copy
- Chris 33-Copy-Copy
- Demo Instance Def
- Devon-Export-Test1

Close Save

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Delete	Delete Button	Removes an instance definition Group.	Deleting an instance definition Group removes it permanently from the database. This does not remove any instance definitions, only the Group.
Add	Add Button	Create a brand-new instance definition Group.	The user needs to select two or more instance definitions to add to the group.
Edit	Edit Button	Edit an instance definition Group.	Editing an instance definition Group modifies the record stored in the database.
Export	Export button	Export an instance definition Group.	Exporting an instance definition Group triggers a download of the exported instance definition Group to a JSON file.

Permissions

OPERATION	ROLE	DESCRIPTION
Add	InstanceDefinitionCreator	The instance definition creator can add instance definition groups.
Add	InstanceDefinitionAdmin	The instance definition admin can add instance definition groups.
Edit	InstanceDefinitionCreator	The instance definition creator can edit instance definition groups.
Edit	InstanceDefinitionAdmin	The instance definition admin can edit instance definition groups.
Delete	InstanceDefinitionCreator	The instance definition creator can delete instance definition groups.
Delete	InstanceDefinitionAdmin	The instance definition admin can delete instance definition groups.
Export	InstanceDefinitionViewer	The instance definition viewer can export instance definition groups.
Export	InstanceDefinitionCreator	The instance definition creator can export instance definition groups.
Export	InstanceDefinitionAdmin	The instance definition admin can export instance definition groups.

How to use the Token Search

This section will describe what the token search is, its purpose, how to use it, and how tokens are cached and displayed in the token search.

Tokens

In RPS, tokens are used as placeholders for values which populate when the instance definition is invoked. Tokens allow items and instance definitions to be more dynamic in where values are populated during invocation.

Tokens built into RPS

There are a few tokens built into RPS that will not require a value to be passed in when invoking the instance definition. Typically, these built-in tokens are used to make an item's "Entity Name" unique, but they can be used wherever the user needs.

FIELDS	DEFINITION
Entity ID	this is the ID of the item
Parent Name	this is the name of the parent item

Token Search

The token search allows the user to browse through cached tokens, eliminating the need to manually type them in each time. The tokens displayed in the token search are updated once every 60 seconds with new tokens. The token search works by checking newly created items to see if there are tokens in an item's fields or properties. It adds any new tokens to the token search. Additionally, if a user has deleted a token from the token picker and saved an item that contains that token, that token will be added back to the cached token list.

Bring up the Token Search

1. When creating/editing an instance definition item (See [Creating or Editing Instance Definition Items](#) for instructions), hover over the "Entity Name" field box, and a search icon appears.



Figure 1 - Token Search Icon

1. Click on the search icon and the Token Search will pop up and appear in the middle of the browser's window. This window will contain a search box which will enable the user to find cached tokens.

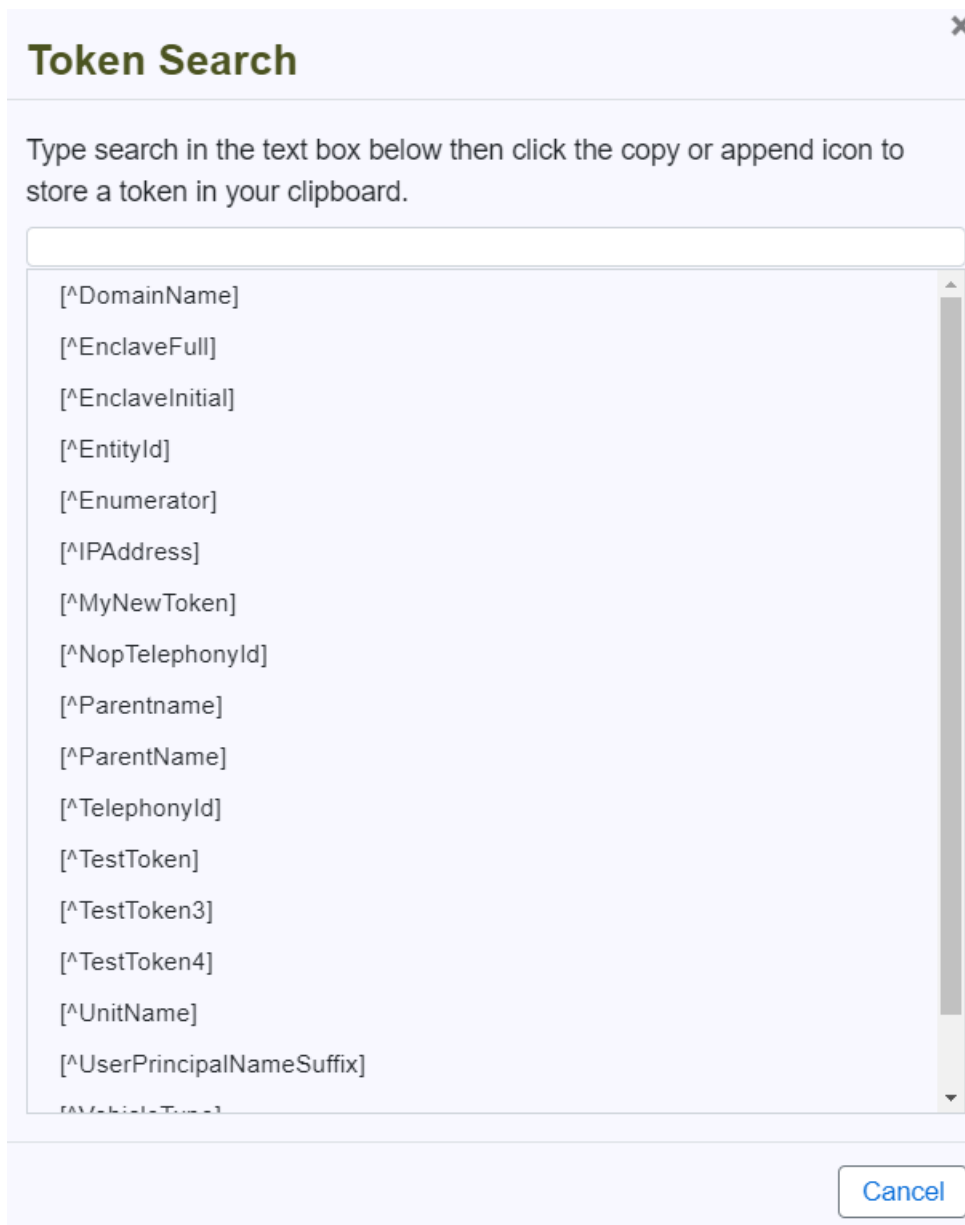


Figure 2 - Token Search Window

Use the Token Search

1. The token search window displays the cached tokens, which allows the user to search for the desired token using the search bar.

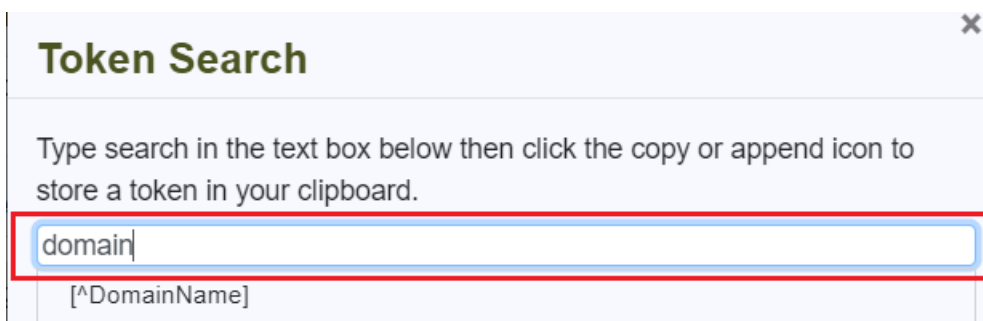


Figure 1 - Search Function

2. The user can select a token by hovering over the desired token, which then display 3 icons (append, copy, and delete) to the right of the token name. From this step, the user can perform the copy, append, and delete tasks. The following sections explains how to use each of the icons and their purpose:

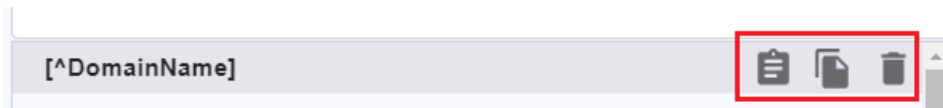


Figure 2 - Token Search Icons

Copy Token

This action copies a single token that the user chooses, and provides enough data to fill out the value for the "Entity Name" or "Property" value.

1. Click on the Copy icon of the wanted token.

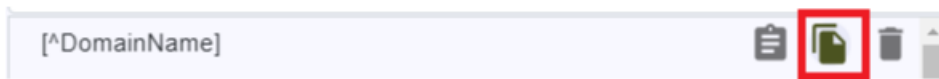


Figure 1 - Copy Icon

2. Then the "Token Search" window will close.
3. Paste the copied token into the field.

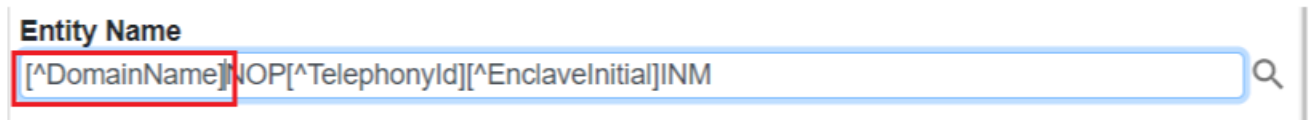


Figure 2 - Successful Copy

Append Token

If a single token does not provide enough data to fill out the "Entity Name" or "Property" values, users may append, or add, an additional token to the instance definition.

1. Append icon of the wanted token



Figure 1 - Append Icon

2. Click on the Append icon of the wanted token. This adds the newly selected token to the previous one in the clipboard. A success message appears in the bottom right of the window.

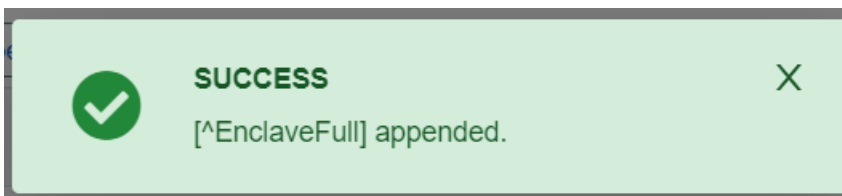


Figure 2 - Successful Append Window

3. Repeat step 2 until user has appended all wanted tokens

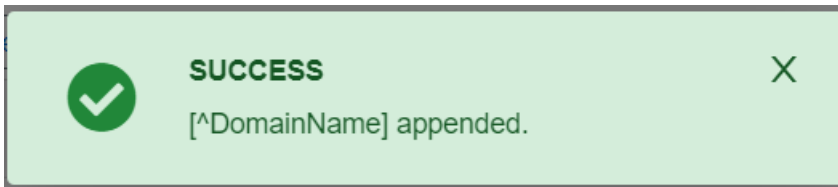


Figure 3 - Successful Append Window

4. Close the Search Box by clicking on the "close" icon or on the "cancel" button.

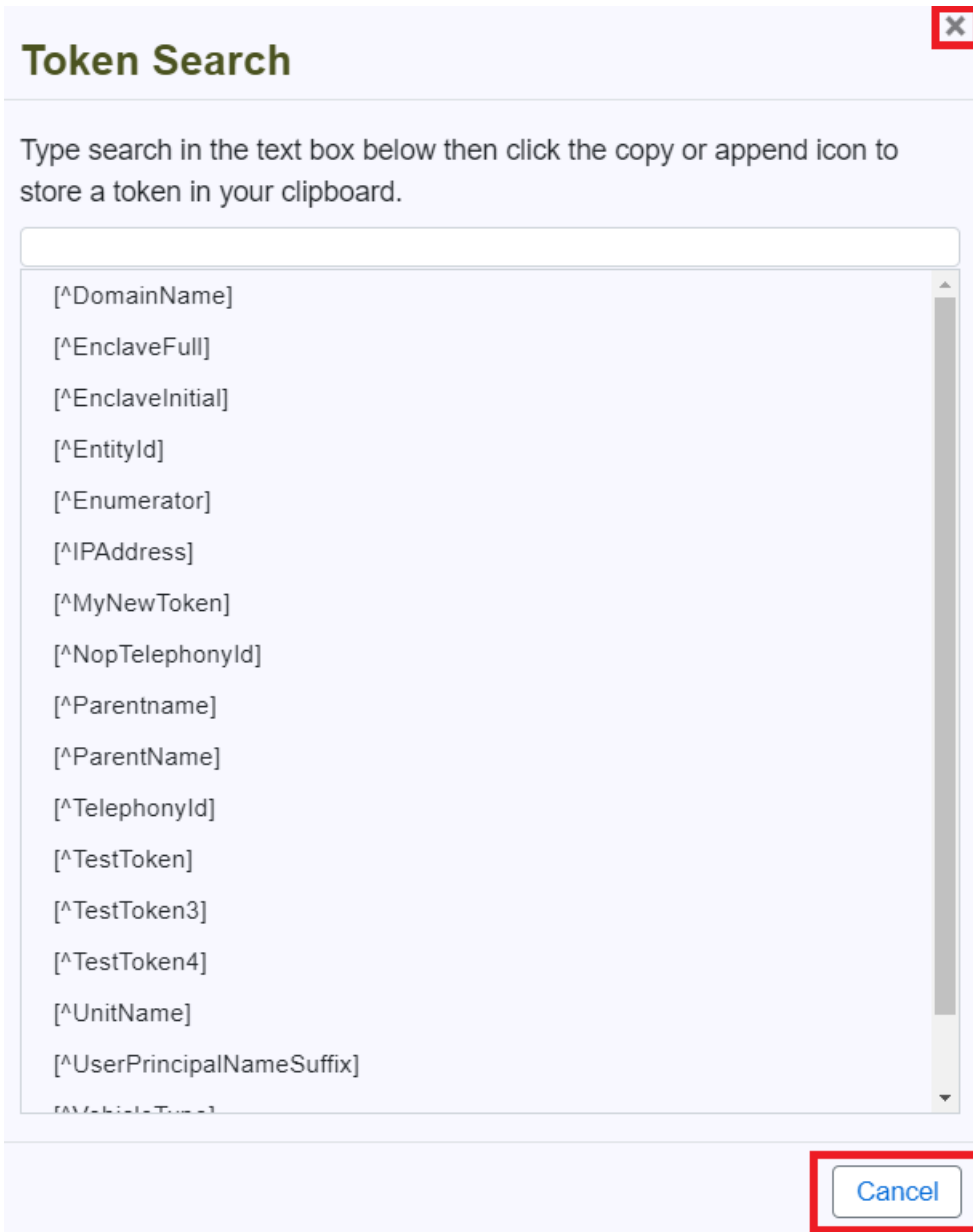


Figure 4 - Exiting the Token Search

5. Paste the copied token into the field using CTRL + V or by right-clicking mouse and selecting "paste" in order to paste. Users see the appended tokens inside the field as show in the following figure:

```
! [Completed Token Appended 5]
(../../../../Images/REACTR/InstanceDefinitions/InstanceDefinitionsUserGuide/TokenPicker_Append_Successful_EntityName.png)
```

Figure 5 - Successful Append in Entity Name

Delete Token

NOTE

This action does not remove the token from any existing instance definitions or instance definition items.

1. Click on the "Delete" icon to the right of the token.

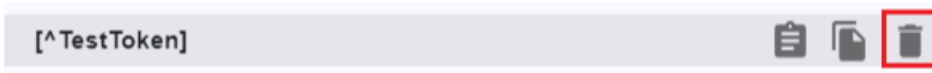


Figure 1 - Selecting token to delete

2. A Delete confirmation window appears in the center of the window. Click on the "Delete" button.

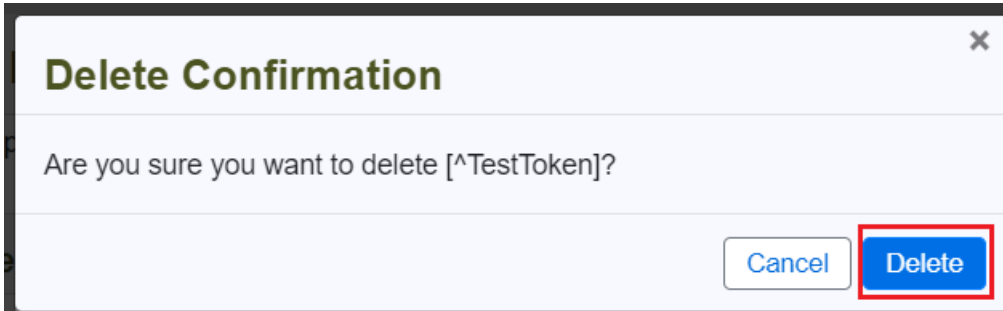


Figure 2 - Delete Confirmation box

3. A confirmation of successful delete appears in the bottom right of the window.

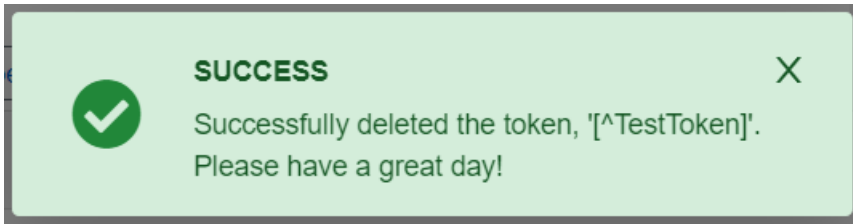


Figure 3 - Successful Deletion confirmation

4. The deleted token is no longer shown in the token search.

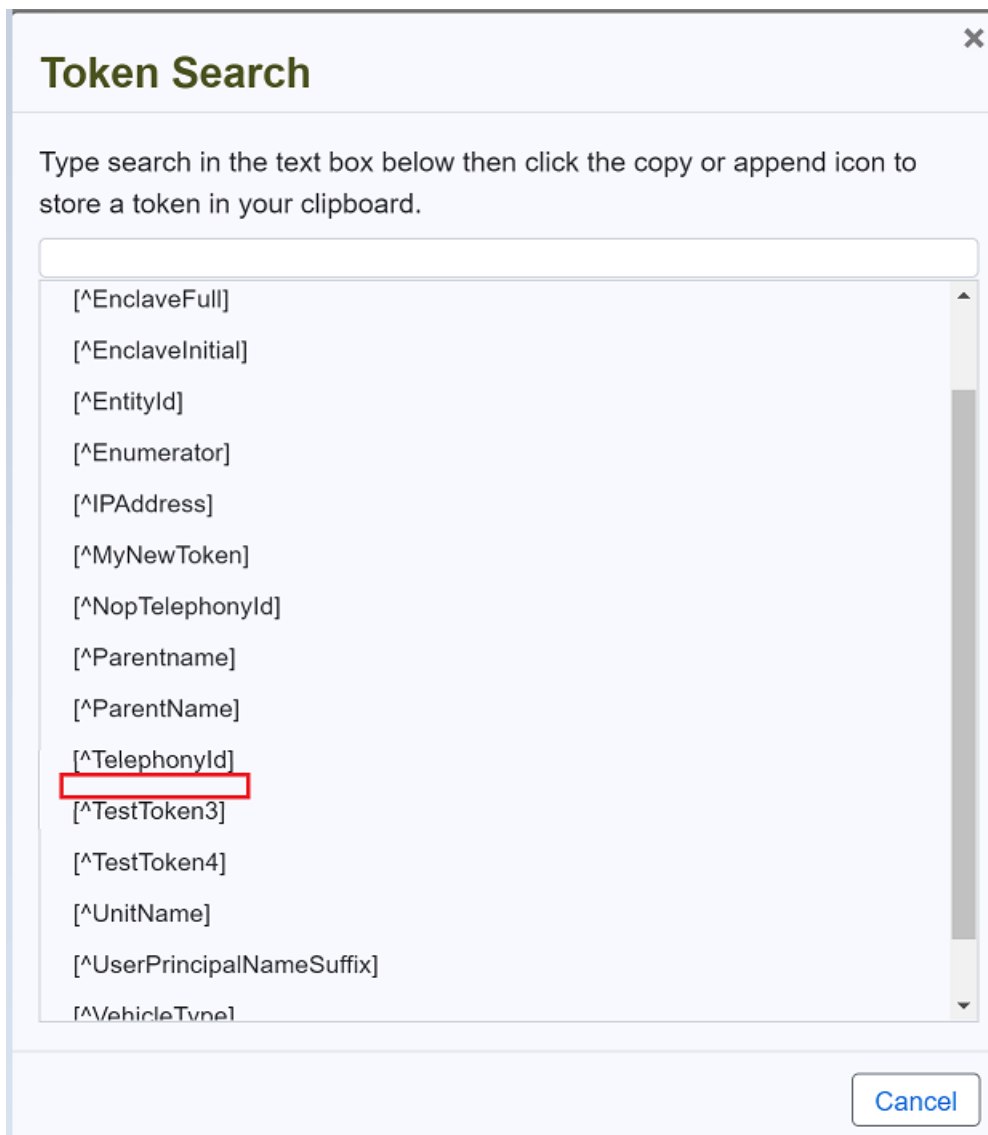


Figure 4 - Successful Deletion in Token Search

Bulk Replace Tokens

Tokens can be bulk replaced in instance definition items. By using the Replace Tokens functionality, the user can specify the old token to replace and the new token that should replace it. Token search can be used for the *Old Token* and *New Token* fields.

WARNING

This will update the token in **all** items that contain the old token.

The steps necessary to bulk replace tokens are described below:

1. Navigate to **Instance Definitions > View All Instance Definition Items**
2. Click on the **Replace Tokens** button.

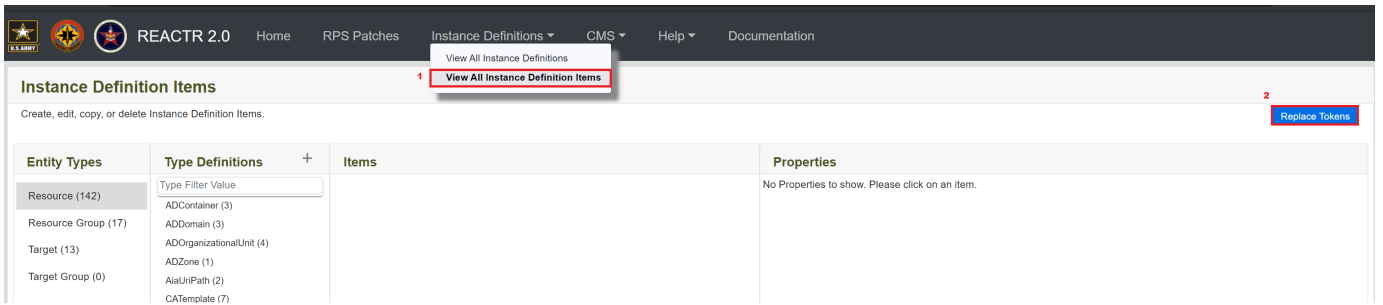


Figure 1 - Instance Definition Item Replace Tokens

3. Specify the value of the *Old Token* that you want to replace and the *New Token* that you want to replace it with.
4. Specify if the tokens should be replaced in the entity name and/or the properties (both checked by default).
5. Click on the **Replace** button to bulk replace all matching tokens.

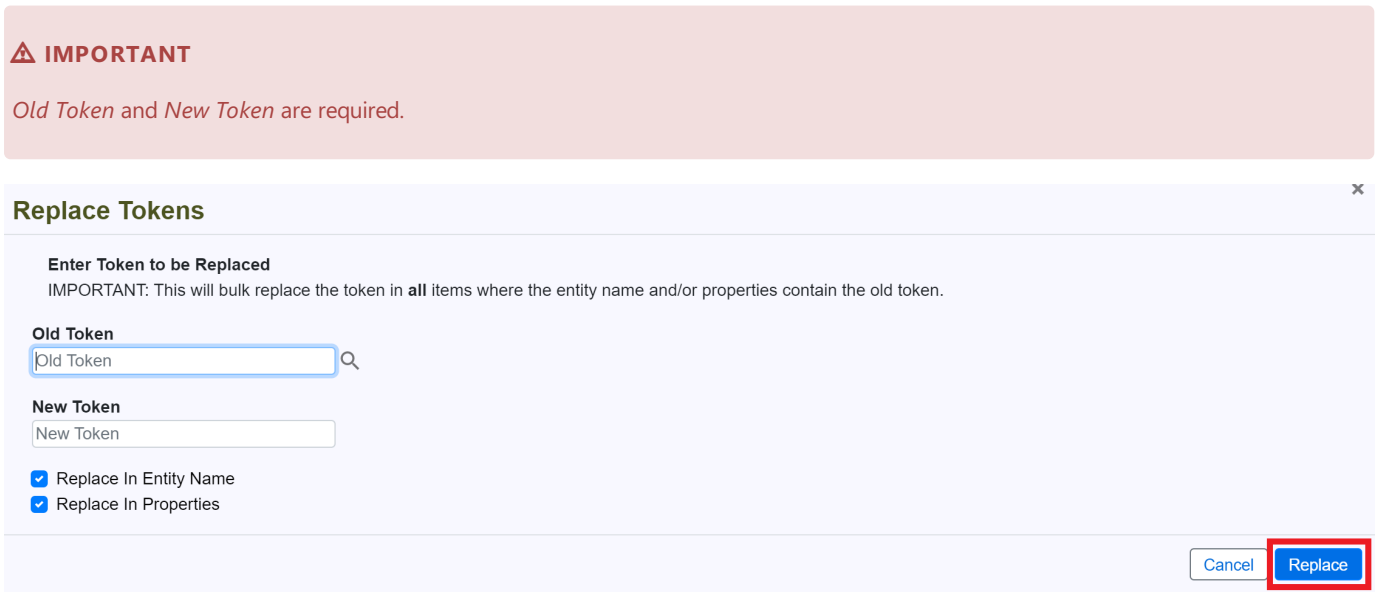


Figure 2 - Instance Definition Item Replace Tokens Screen

6. A list will be shown of all the items that were updated during the bulk replace.

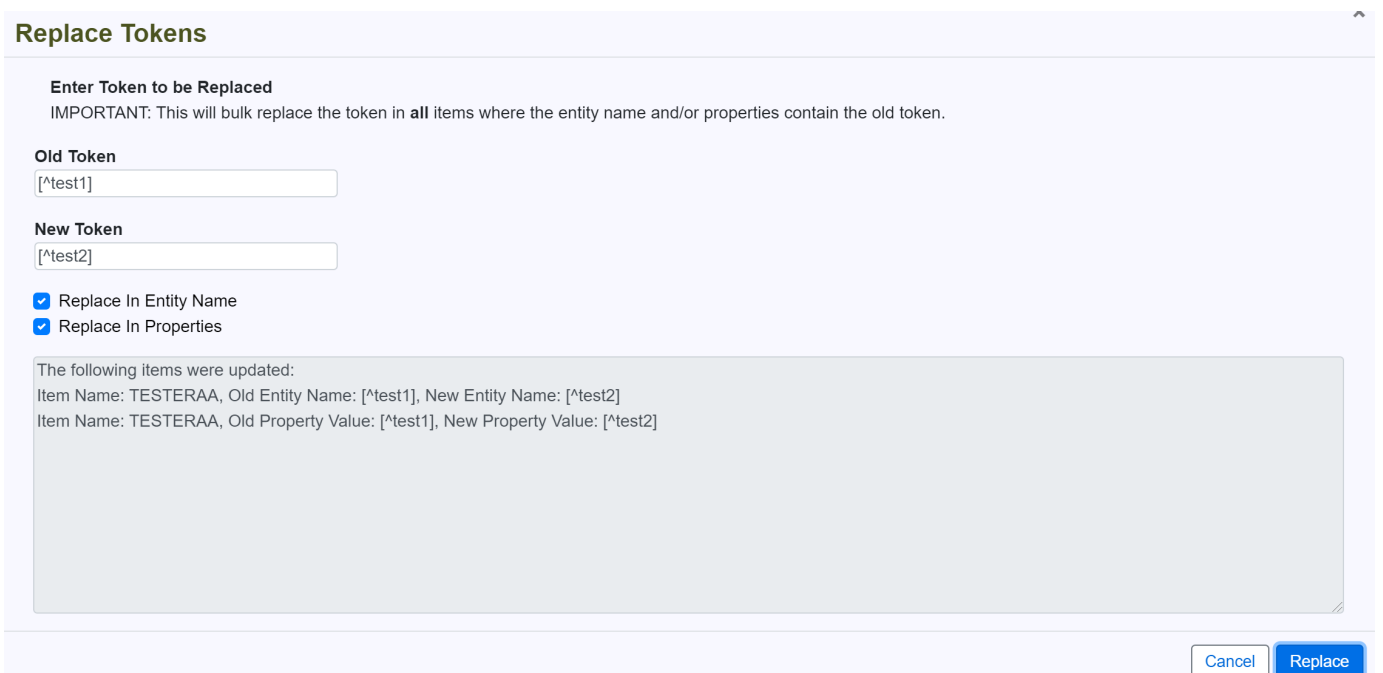


Figure 3 - Instance Definition Item Replace Tokens Result

Operations

OPERATION NAME	TRIGGER	PURPOSE	ADDITIONAL COMMENTS
Replace Tokens	Replace Tokens Button	Bulk replaces the old token with the new token in the entity name and/or properties.	This will affect all items that contain the old token.

Permissions

OPERATION	ROLE	DESCRIPTION
Replace Tokens	InstanceDefinitionCreator	The instance definition creator has the ability to bulk replace tokens.
Replace Tokens	InstanceDefinitionAdmin	The instance definition admin has the ability to bulk replace tokens.

How to use Special Characters

Certain special characters are supported when entered in the property values for Instance Definition Items or Nodes.

Supported Special Characters

SPECIAL CHARACTER	GETS REPLACED WITH	DESCRIPTION
{linefeed}	\n	A line feed means moving one line forward
{newline}	\r\n	A new line in windows is the combination of \r\n. In Linux the equivalent is usually just \n
{carriagereturn}	\r	A carriage return means moving the cursor to the beginning of the line

How to Add a Special Character to a Property Value

1. To edit an Instance Definition Item and its properties or an Instance Definition's Node and its properties follow the instructions in the sections [Edit Instance Definition Items](#) or [Edit an Instance Definition](#).
2. When editing the value of an Instance Definition Item Property or a Node Property, you can add one or more special characters in the property value.
3. When saving the Instance Definition Item or Instance Definition (and its Node) the special characters in the property value will be replaced as specified in the table above.

Scenario: How to Create and Export an Instance Definition and its Items

This section describes an end-to-end scenario of how to create, edit, and export an instance definition and its items. Throughout this scenario there are links to various sections of this document that guides you how to accomplish each step of the workflow.

1. The first step in creating an instance definition is to create the individual items that will belong to the instance definition.
 1. Items are reusable between multiple instance definitions and can be 'templated' by using tokens (see: [Add Tokens to a Property](#)) in any properties that may differ between instance definitions.
 2. Once an item is ready to be used in an instance definition, its item Approved should be set to 'Approved' so that further changes to it are prevented (admins can toggle this back to 'Not Approved').
 3. To create instance definition items see: [Create or Edit Instance Definitions](#)
 4. To delete instance definition items see: [Delete an Instance Definition Item](#)
 1. Since items can be reused between multiple instance definitions, you are not able to delete an item that is referenced in one or more instance definitions. These references must first be removed in order to delete the

item.

2. After the instance definition items have all been created, then you can start creating the instance definition which is the template/scaffold/structure for how the individual items are used.
 1. To create instance definitions see: [Create or Edit Instance Definitions](#)
 2. To delete instance definitions see: [Delete an Instance Definition](#)
 3. To duplicate instance definitions see: [Copy Instance Definitions](#)
 1. This copies only the instance definition and its references and associations. This does not copy the actual individual items.
3. Once the instance definition has been created, then you can start configuring/building it by adding references to items.
 1. To add or remove a reference to an item see: [Add or Remove Items in an Instance Definition](#)
 2. To replace a reference to an item see: [Replace Items in an Instance Definition](#)
 3. To configure/build referenced items see: [Add or Remove Items From a Referenced Item](#)
 1. By configuring referenced items you can add children underneath the referenced item.
4. When all references to items have been added to the instance definition, you can continue configuring/building it by adding associations between referenced items.
 1. To configure/build referenced items see: [Add or Remove Items From a Referenced Item](#)
5. The last step before exporting is to verify that all your data is correct by reviewing the:
 1. Properties of instance definition
 2. Properties of instance definition items
 3. References to instance definition items
 4. Associations between referenced instance definition items
6. The final step is to export the instance definition to a JSON file.
 1. To export an instance definition see: [Export an Instance Definition](#)

Annotated Screenshot Guides

Instance Definitions Listing Annotated Guide

Instance Definitions

Purpose: List the existing Instance Definitions and provide actions to configure, copy, edit, delete, or export them. Also, this page provides an action to add new Instance Definitions

Status of each Instance Definition. Inactive Instance Definitions are not able to be edited or configured.

The screenshot shows a table with the following structure:

Name	Node Name	Status	Actions
JNN-NGP-InstanceDefinition	JNN-NGP-Node	Inactive	[Search] [Edit] [Delete] [Export]

Annotations in the screenshot:

- A red box highlights the '+' button in the Actions column, with a note: "This button provides an action to create new Instance Definitions".
- Red boxes highlight the [Search], [Edit], [Delete], and [Export] buttons in the Actions column, with a note: "These buttons provide actions to configure, copy, edit, delete, or export specific Instance Definitions".

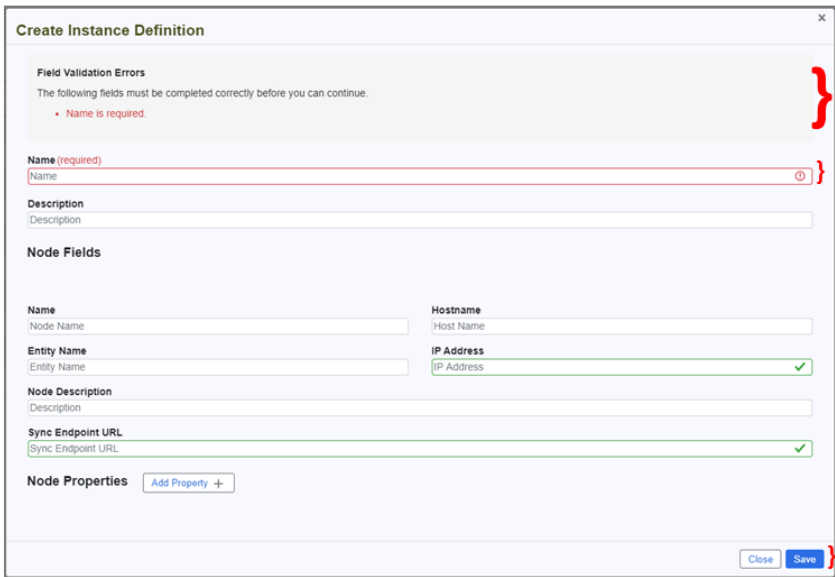
This button provides an action to create new Instance Definitions

These buttons provide actions to configure, copy, edit, delete, or export specific Instance Definitions.

Create or Edit Instance Definition Annotated Guide

Create/Edit Instance Definition

Purpose: Provides a way to create or edit an Instance Definition and its Node.



The screenshot shows a form titled "Create Instance Definition" with several sections and fields. Red annotations highlight specific features:

- Field Validation Errors:** A message at the top states "The following fields must be completed correctly before you can continue." with a red bullet point: "Name is required."
- Name (required):** A text input field for "Name" is highlighted in red, indicating it is a required field that is currently empty.
- Description:** A text input field for "Description".
- Node Fields:** A section containing several fields:
 - Name:** Node Name
 - Entity Name:** Entity Name
 - Node Description:** Description
 - Sync Endpoint URL:** Sync Endpoint URL (with a green checkmark)
 - Hostname:** Host Name
 - IP Address:** IP Address (with a green checkmark)
- Node Properties:** A section with an "Add Property +" button.
- Buttons:** "Close" and "Save" buttons at the bottom right.

Annotations on the right side of the form:

- "Validation errors will be displayed at the top in red." (pointing to the error message)
- "Required fields that are not filled in are highlighted in red." (pointing to the red border around the Name field)
- "Save button will save any changes made to the Instance Definition." (pointing to the Save button)

Annotations on the left side of the form:

- "Node fields are used to describe the Node of the Instance Definition." (pointing to the Node Fields section)
- "Node properties allow you to add extra information to the Node by adding one or more key/value pairs (properties)." (pointing to the Node Properties section)

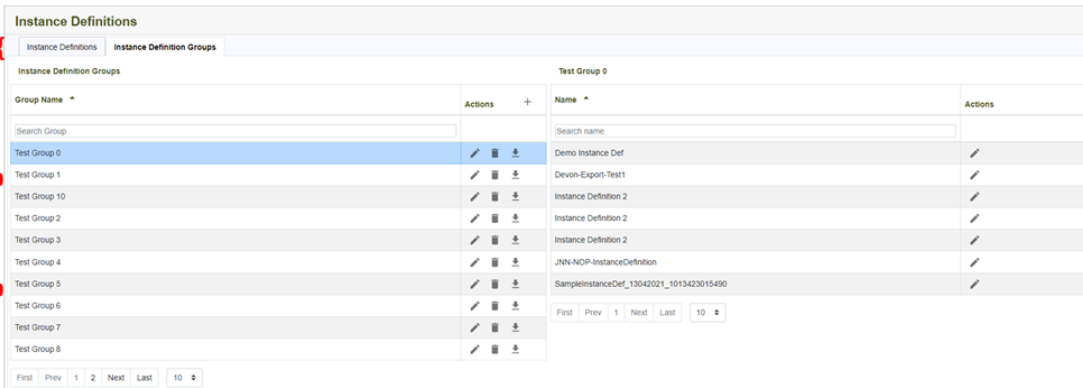
Instance Definition Groups Annotated Guide

Instance Definition Groups

Purpose: Lists the existing Instance Definition Groups and provide actions to create, edit, delete, or export them.

The tabs provide a way to switch between viewing the Instance Definitions or the Groups.

List of Instance Definition Groups with actions to edit, delete, or export each. Also, each group name can be clicked on to select it and to view the Instance Definitions within it on the right side of the screen.



The screenshot shows a web interface with two tabs: "Instance Definitions" and "Instance Definition Groups". The "Instance Definition Groups" tab is active, displaying a table of groups. The "Test Group 0" group is selected, and its details are shown on the right side of the screen.

Group Name *	Actions +	Name *	Actions
Search Group		Search name	
Test Group 0	[edit] [delete] [export]	Demo Instance Def	[edit]
Test Group 1	[edit] [delete] [export]	Devon-Export-Test1	[edit]
Test Group 10	[edit] [delete] [export]	Instance Definition 2	[edit]
Test Group 2	[edit] [delete] [export]	Instance Definition 2	[edit]
Test Group 3	[edit] [delete] [export]	Instance Definition 2	[edit]
Test Group 4	[edit] [delete] [export]	JNN-NOP-InstanceDefinition	[edit]
Test Group 5	[edit] [delete] [export]	SampleInstanceDef_13042021_1013423015490	[edit]
Test Group 6	[edit] [delete] [export]		
Test Group 7	[edit] [delete] [export]		
Test Group 8	[edit] [delete] [export]		

Annotations on the right side of the screenshot:

- "List of Instance Definitions with action to edit each. These are the Instance Definitions within the selected group." (pointing to the list of definitions in the right-hand pane)

Create or Edit Instance Definition Group Annotated Guide

Create/Edit Instance Definition Group

Purpose: Provides a way to create or edit an Instance Definition Group.

Name of the group. {

List of Instance Definitions that you can checkmark to be included in the group. {

Save button will save any changes made to the group. }

Instance Definition Builder Annotated Guide

Instance Definition Builder

Purpose: Provides a way to configure the Instance Definition and its Items.

Slide out panel that displays a tree view of the Instance Definition

Breadcrumb that indicates where you are in the Instance Definition's hierarchy

Info icon that can be clicked to view the properties of the referenced item.

Actions that allow you to delete, configure, or change out the referenced item

Entity Types are core groupings of items that can exist in an Instance Definition. Every Item is one of these four types.

This list of Type Definitions are the available schemas that are applied to Items. Every item must have one Type Definition that it will use to validate against.

This list of referenced items are items that are already referenced for the selected Type Definition.

Details and Properties of the item for the selected reference.

Add Reference to Instance Definition Annotated Guide

Instance Definition Builder – Add Reference

Purpose: Provides a way to reference Items in an Instance Definition

List of referenced items that can be referenced to the parent reference item or Instance Definition (if at the top/root level)

Select button will save any references that were added or removed to the parent reference item or Instance Definition (if at the top/root level)

Details and Properties of the item for the selected item.

Add Association to Instance Definition Annotated Guide

Instance Definition Builder – Add Association

Purpose: Provides a way to associate referenced Items in an Instance Definition

List of referenced items that can be associated to the parent reference item

Select button will save any associations that were added or removed to the parent reference item

Change a Referenced Item in Instance Definition Annotated Guide

Instance Definition Builder – Change Reference

Purpose: Provides a way to change what Item is being referenced in an Instance Definition

Instance Definition Items Annotated Guide

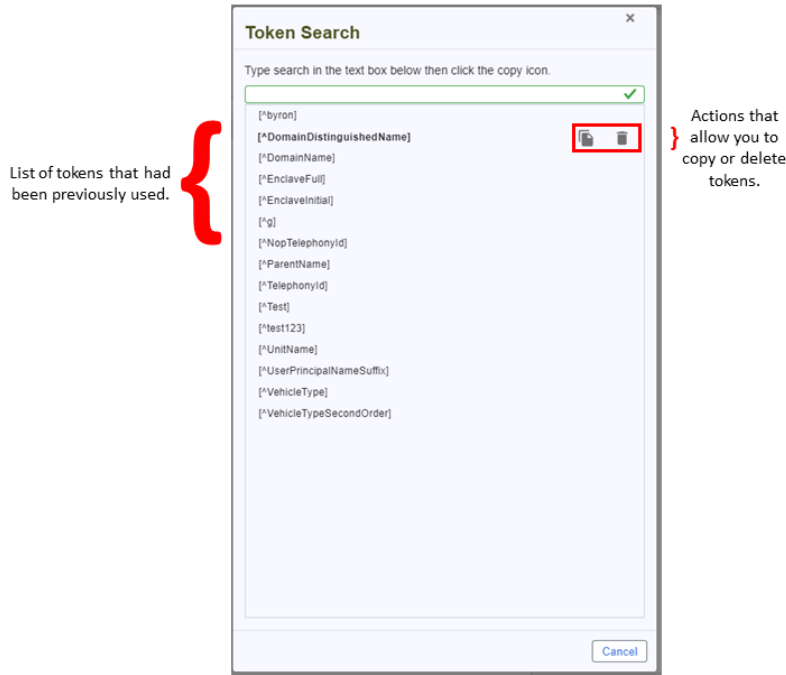
Instance Definition Items

Purpose: Provides a way to create, edit, delete, or copy Instance Definition Items

Token Picker Annotated Guide

Token Picker

Purpose: Provides a way to pick from previously used tokens and reuse them elsewhere.



Glossary

TERM	DEFINITION
RPS	Rapid Provisioning System
REACTR	RPS Enterprise Automation, Configuration and Testing Routines
Instance Definition	The abstraction layer on top of existing RPS Types.
Instance Definition Reference	The assignment of the instance definition to a root resource item, which results in a set of one or more resource items to be run.
Instance Definition Association	The association between referenced items results in an assignment between resource items or resource groups to either target items or target groups. These assignments allow one-to-one, many-to-one, one-to-many, and many-to-many relationships.
Instance Definition Item	An item on the abstraction layer on top of existing RPS Types.
Instance Definition Node	An item that will result in association of an endpoint node with target items that are created when an instance definition is invoked.
Instance Definition Group	A collection of two or more instance definitions. Groups provide the ability to export multiple instance definitions to a single JSON file.