SIM v5

Release

Sep 15, 2017

Contents

Note: Documentation is still in development process. Please do not hesitate to contact us on support@silvermonkey.net for further information.

CHAPTER 1

Welcome

This document is meant to be a source for all information regarding the administration and installation of the new OPS module.

This document was generated on 2017-09-15 at 08:37.

Contents:

Introduction

In this article:

OPS Module overview

The OPS (Operations) Module enables users to view the current state of different domain entities (such as Computers, Applications, Users etc.) and allows to invoke actions (such as WakeOnLan, Install Application etc.) on these entities.

Access to OPS is controled by the SIM access control system (domain groups are mapped to SIM roles).

The OPS Module consists of three seperate services: ops-auth, ops-api and ops-web. The end-user will only deal with ops-web (the frontend), whereas administrators also need to configure ops-auth and ops-api (the backend). The OPS Module actions that can be invoked for the domain entities are delegated to v5/v6 Forms. Therefore the OPS Module has a dependency on v5/v6.

ops-auth

The ops-auth module (or simpler just auth module) is a service to retrieve OPS user roles (not Windows or Azure AD roles).

> **Warning:** Note. The ops-auth module will be replaced in the near future

ops-api

The ops-api module is a service that provides access to one or more sql tables as well as defines the views and actions that will be shown in its clients (for example the ops-web). Access to the data, the views and actions is determined by SIM roles.

ops-web

ops-web is the frontend that end-users will use to view entities and invoke actions on them. The views and actions that are available are configured in ops-api.

Requirements

Serverside

Name	Technologie	Operating System	Webserver	.NET Framework
ops-api	ASP.NET	Windows 7, Windows Server 2012 (or higher)	IIS 8 (or higher)	4.5.2 (or higher)
ops-auth	ASP.NET Core	Windows 7, Windows Server 2012 (or higher)	IIS 8 (or higher)	4.5.2 (or higher)

ops-auth | ASP.NET Core | Windows 7, Windows Server 2012 (or higher) | IIS 8 (or higher) | 4.5.2 (or higher) |

Clientside

any modern browser SHOULD work.

Installation

In this article:

ops-auth

The module needs to be installed as an IIS application for a website. The websites url needs to be known by SIM before compiling the application.

Applicationname		auth
Applicationpool	.NET CLR Version	v4.0.30319
	Managed pipeline mode	Integrated
	Identity	Custom account with read access to the database
Authentication		Only anonymous authentication enabled

ops-api

The module needs to be installed as an IIS application for a website. The websites url needs to be known by SIM before compiling the application.

Applicationname		ops-api
Applicationpool	.NET CLR Version	No Managed Code
	Managed pipeline mode	Integrated
	Identity	Custom account with read access to the database
Authentication		Only anonymous authentication enabled

IIS Modules

The AspNetCoreModule module needs to be activated for this application.

ops-web

The ops-web can be hosted in any webserver. If it is hosted in IIS the following settings apply:

Applicationname		ops or ops-web or ops-webapp
Applicationpool	.NET CLR Version	v4.0.30319
	Managed pipeline mode	Integrated
	Identity	Applicationpool Identity
Authentication		Only anonymous authentication enabled

Database Setup

In this article:

ops-auth

The module requires an MS SQL Server and database that provides the mapping between Windows domain roles and OPS roles. The connection to the database can be specified in the modules Web.config DefaultConnectionString node. The connection strings property "integrated security" SHOULD have the value "true", the "provider name" property SHOULD have the value "System.Data.EntityClient". Under the node appSettings the value of "DomainName" MUST have the name of the windows domain.

example Web.config

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <appSettings>
        <add key="DomainName" value="phatconsulting.group" />
    </appSettings>
    <connectionStrings>
        <add name="DefaultConnectionString" connectionString="data source=simsrv042;
↪initial catalog=SIM_OPS_R042;integrated security=True;MultipleActiveResultSets=True;
↪" providerName="System.Data.EntityClient" />
    </connectionStrings>
    <system.web>
        <authentication mode="Windows" />
        <compilation targetFramework="4.6.1">
        <assemblies>
            <add assembly="System.Security, Version=4.0.0.0, Culture=neutral,␣
↪PublicKeyToken=B03F5F7F11D50A3A" />
            <add assembly="System.DirectoryServices, Version=4.0.0.0, Culture=neutral,
↪ PublicKeyToken=B03F5F7F11D50A3A" />
            <add assembly="System.DirectoryServices.AccountManagement, Version=4.0.0.
↪0, Culture=neutral, PublicKeyToken=B77A5C561934E089" />
        </assemblies>
        </compilation>
        <httpRuntime targetFramework="4.5.2" />
    </system.web>
</configuration>
<!--ProjectGuid: b03c2d11-e5ac-4242-a5c2-862a3787e00a-->
```

The auth module expects a Table named "Role_Group" with columns "RoleId" (uniqueidentifier, not null) and "GroupName" (varchar, not null) in the given database. The Groupname is the name of a windows domain group. The RoleId is the id of a SIM/OPS role. The corresponding Role table is not used by the auth module and therefore CAN be absent.

ops-api

The ops-api module expects following tables and relationships:

Role table:

Column	Type
Id	PK, uniqueidentifier, not null
Name	Name, varchar(1000), not null

any number of **Item** tables: (name can be chosen arbitrarily)

Column	Type
Id	PK, uniqueidentifier, not null
ItemType	varchar(15), not null

foreach **Item** table there MUST be exactly one corresponding **Item_Role** table (name can be chosen arbitrarily), that defines a many-to-many relationship between the corresponding **Item** table and the **Role** table:

Column	Type
RoleId	FK Role(Id), uniqueidentifier, not null
ItemId	FK **Item**(Id), uniqueidentifier, not null

Note: database conventions SIM (usually) uses

- singular for table names (for example "Application" instead of "Applications")
- "Id" as the name for the PRIMARY KEY
- The GUID/uniqueidentifier type for the PRIMARY KEY column
- The names of the involved tables seperated by an underscore in a many to many relationship (for example "Computer_Role")
- Tablename + "Id" for FOREIGN KEYS (for example "RoleId")

Config schema

This document describes the schema of the config.json file. The config.json file is used to configure the available views and actions for the web-app.

Root object

Name	Type	Description	Required
Sites	array<SiteConfig>	A list of Sites	required
Panels	array<PanelConfig>	A list of Panels	required
Filters	array<FilterConfig>	A list of Filters	required
ContextMenus	array<ContextMenu>	A list of ContextMenus	required
DndMenus	array<DnDMenu>	A list of DndMenus	required
Menus	array<Menu>	A list of Menus	required

SiteConfig

Name	Type	Description	Required
Id	integer		required
DisplayName	string	The name that will be displayed to users	required
PanelIds	array<integer>	A Site can display several Panels that are specified here	required
RoleNames	RoleNames	A Site can be shown to only authorized users, which can be described here	

RoleNames

Name	Type	Description	Required
Values	array<string>	The actual values. Roles can end with a wildcard, to match more than one role at once. The symbol * (asterisk) is used. Example: Admin* would match any role that begins with **Admin**, so Admin, AdminHamburg and AdminBerlin would match the rule. GlobalAdmin however would not match.	required

PanelConfig

Name	Type	Description	Required
Id	integer	The id property	required
SearchField	string	The property name (usually the name of the column) that should be used for searching	required
Columns	array<Column>	The columns that should be displayed in the table. Don't forget that every item should have the properties **id** and **itemType** even if they are not displayed	required
FilterIds	array<integer>	Specifies the available Filters	required
Table	string	The name of the table to display	required
PermissionTable	string	The many-to-many intermediate table between the table specified in the Table property and the Role table	required
DefaultCondition	string	A sql expression to filter the items	

Column

Name	Type	Description	Required
FieldName	string	The name of the property (usually a database column)	required
DisplayName	string	The label that should be displayed in the UI	required

FilterConfig

Name	Type	Description	Required
Id	integer	Id must be unique among all filters	required
DisplayName	string	The name that will be displayed in the UI	required
ActionsMenuId	integer	Every filter is bound to a specific ActionMenu	required
RoleNames	RoleNames	A filter can be made available to specific roles. If this property is not set, the filter will be available for all users	
Condition	string	An SQL condition that is used as part of a where clause when querying the database table	

ContextMenu

Name	Type	Description	Required
ItemType	string	TODO	required
MenuId	integer	The id of the corresponding Menu	required

DnDMenu

Name	Type	Description	Required
ItemTypes	array<string>	A combination (a list) of itemtypes, for which this Drag-and-Drop-Menu will be available	required
MenuId	integer	The id of the corresponding Menu	required

Menu

Name	Type	Description	Required
Id	integer		required
Name	string		required
MenuItems	array<MenuItem>	A list of actions that are bound to this menu	required

MenuItem

Name	Type	Description	Required
DisplayName	string	Will be used in the UI	required
UrlTemplate	string	UrlTemplate is used to create a url at runtime with specified parameters from items. Parameters are extracted from the items contextual to the menu. They should have this format: **{[itemType].[column]}** where **itemType** is the mandatory item type that all items should have as a property. Example: **https://v6.com/install?idapp={app.id}&idcomp={computer.id}**.	required
RoleNames	RoleNames	A MenuItem can be restricted to users that belong to certain Roles	

config.json sample file

```
{
  "sites": [
    {
      "id": 1,
      "displayName": "Everything",
      "panelIds": [ 2 ],
      "roleNames": { "values": [ "Admin", "User" ] }
    },
    {
      "id": 2,
      "displayName": "Restricted",
```

```
      "panelIds": [ 3 ],
      "roleNames": { "values": [ "Admin", "RoleWithWildCard*" ] }
    }
  ],
  "panels": [
    {
      "id": 2,
      "table": "[dbo].[Item]",
      "permissionTable": "[dbo].[ItemsRoles]",
      "columns": [
        {
          // Name of the db column for this field.
          "fieldName": "name",
          // Name displayed on frontend for this field.
          "displayName": "Computer Name"
        },
        {
          // Name of the db column for this field.
          "fieldName": "description",
          // Name displayed on frontend for this field.
          "displayName": "Computer Description"
        },
        {
          // Name of the db column for this field.
          "fieldName": "dn",
          // Name displayed on frontend for this field.
          "displayName": "dn"
        },
        {
          // Name of the db column for this field.
          "fieldName": "domainAlias",
          // Name displayed on frontend for this field.
          "displayName": "Alias"
        }
      ],
      "searchField": "name",
      "defaultCondition": "",
      "filterIds": [ 0, 3 ]
    },
    {
      "id": 3,
      "table": "[dbo].[Item]",
      "permissionTable": "[dbo].[ItemsRoles]",
      "columns": [
        {
          "fieldName": "name",
          "displayName": "Computer Name"
        },
        {
          "fieldName": "description",
          "displayName": "Computer Description"
        },
        {
          "fieldName": "dn",
          "displayName": "dn"
        },
        {
          "fieldName": "domainAlias",
```

```
          "displayName": "Alias"
        }
      ],
      "searchField": "name",
      "defaultCondition": "",
      "filterIds": [ 3 ]
    }
  ],
  "filters": [
    {
      "id": 0,
      "displayName": "All",
      "actionsMenuId": 0
    },
    {
      "id": 3,
      "displayName": "With description 2",
      "condition": "Name IS NOT NULL",
      "actionsMenuId": 7
    }
  ],
  "contextMenus": [
    {
      "itemType": "computer",
      "menuId": 1
    },
    {
      "itemType": "app",
      "menuId": 3
    }
  ],
  "dndMenus": [
    {
      "itemTypes": [ "computer", "app" ],
      "menuId": 4
    },
    {
      "itemTypes": [ "computer", "appPrd" ],
      "menuId": 9
    }
  ],
  "menus": [
    {
      "id": 0,
      "name": "A S1 P1",
      "menuItems": [
        {
          "displayName": "Add computer",
          "urlTemplate": "http: //v6.com/add_computer"
        }
      ]
    },
    {
      "id": 1,
      "name": "C S1 P1",
      "menuItems": [
        {
          "displayName": "Delete computer",
```

```
          "urlTemplate": "http: //v6.com/delete_computer/:id",
          "roleNames": { "values": [ "RoleWithWildCard*", "Admin" ] }
        },
        {
          "displayName": "Edit computer",
          "urlTemplate": "http: //v6.com/edit_computer/:id"
        }
      ]
    }
  ]
}
```

Changelog

Version	TicketId	Product	Description
0.1.0	None	OPS	alpha version

Support

If you have further questions regarding our products or the documentation contact us:

- Tel. : +49 40 - 226 383 160
- E-Mail : Support@SilverMonkey.net

If you need general Information about our Products visit: http://www.SilverMonkey.net