

v1.2

Currently `system.sw.os` item key is supported only for Linux OS. It is required that we add support for this key also on Windows OS, as well as, add a new item key `system.sw.os.get` for Linux and Windows that would yield json with system version information.

Use Cases

Case 1: User wants to get system version information for Windows OS.

1. User created new item of type Zabbix agent or Zabbix agent (active).
2. Set item key to `system.sw.os`.
3. Agent now reports OS version of the host.

What's affected:

- Agent
- Agent 2
- Frontend
- Templates
- Documentation

JSON format of `system.sw.os.get []`

For Windows OS (depending on the system version) we can collect following information:

- "os_type" - type of operating system
- "product_name" - Windows OS name
- "architecture" - Targeted processor architecture
- "build_number" - Build number of the OS
- "build_revision" - Revision of the current build of the OS (win 7 and up)
- "build" - String representation of build and revision
- "edition" - Edition of the OS (win 7 and up)
- "composition" - Additional edition name of the OS (version ≥ 6.3)
- "display_version" - Display name of the version (version ≥ 6.3)
- "sp_version" - Service pack version (version < 6.3)
- "sp_build" - Service pack build number (version < 6.3)
- "version" - Version number (6.3 after win 8)
- "version_pretty" - Text representation of the data gathered
- "version_full" - OS name + BuildLab (BuildLabEx for win7and up) + Build

```
{
  "os_type": "windows",
  "product_name": "Windows 11 Pro",
  "architecture": "x86_64",
  "build_number": "22621",
  "build_revision": "674",
  "build": "22621.674",
  "edition": "Professional",
  "composition": "Enterprise",
  "display_version": "22H2",
  "sp_version": "Service Pack 1",
  "sp_build": "1130",
  "version": "6.1",
```

```

    "version_pretty": "Windows 7 Ultimate Build 7601.21214 Service Pack 1",
    "version_full": "Windows 7 Ultimate
7601.24214.amd64fre.win7sp1_Idr_escrow.180801-1700 Build 7601.24214"
}

```

For Linux OS we can collect following information:

- "os_type" - type of operating system
- "product_name" - Name of the distribution
- "architecture" - Targeted processor architecture
- "kernel_major" - major version of the Linux kernel
- "kernel_minor" - minor version of the Linux kernel
- "kernel_patch" - patch version of the Linux kernel
- "kernel" - Kernel version
- "version_pretty" - Text representation of the data gathered
- "version_full" - Full system identifier from /proc/version

```

{
    "os_type": "linux",
    "product_name": "AlmaLinux release 8.6 (Sky Tiger)",
    "architecture": "x86_64",
    "kernel_major": "4",
    "kernel_minor": "18",
    "kernel_patch": "0",
    "kernel": "4.18.0-372.26.1.el8_6.x86_64",
    "version_pretty": "AlmaLinux release 8.6 (Sky Tiger) x86_64 4.18.0-
372.26.1.el8_6.x86_64",
    "version_full": "Linux version 4.18.0-372.26.1.el8_6.x86_64
(mockbuild@c96addf3b7c745e3b437a36677d48bfa) (gcc version 8.5.0 20210514 (Red
Hat 8.5.0-10) (GCC)) #1 SMP Tue Sep 13 06:07:14 EDT 2022"
}

```

Notes:

- if any particular key in the registry of Windows OS is missing, corresponding field will be absent in the resulting JSON.
 - field `os_type` is guaranteed to be present and filled
 - field `version_full` is guaranteed to be present, but not guaranteed to be filled
- if reading of some information bit on Linux failed, then corresponding field will be absent in the resulting JSON.
 - field `os_type` is guaranteed to be present and filled
 - field `version_full` is guaranteed to be present, but not guaranteed to be filled
- "os_type" field can have following values (determined during compilation):
 - "windows" for all systems in Windows family
 - "linux" for all systems based on Linux kernel

Agent

Library `zbxsysinfo` gets expanded:

- new key of `system.sw.os` added for windows:
 - making use of function `RtlGetVersion()` it is possible to get windows major and minor versions as well as build number, for older versions this function also provides service pack numbers.

- new key of `system.sw.os.get` added for Linux and Windows:
 - using similar implementation to `system.sw.os` implement JSON item that will provide OS information.

Agent 2

- create new built-in package for windows with `system.sw.os` and `system.sw.os.get` keys
- create new built-in package for linux:
 - implement there new key `system.sw.os.get []`
 - implement Go version of the `system.sw.os` (alleviate dependency on C code)

Frontend

Add new item key of `system.sw.os.get` to list of possible keys for item types:

- Zabbix agent
- Zabbix agent (active)

Documentation

- Now `system.sw.os` key is supported on Windows OS.
- New item `system.sw.os.get` that created JSON with OS information. Supported for Windows and Linux.

Templates

- "Windows by Zabbix agent" and "Windows by Zabbix agent active" templates:
 - new item with key `system.sw.os`
 - new trigger for system version change