**Автоматизована система**

**«Файловий агент АРМ ВЗ»**

Версія 1.0.0

**Технічна документація**

**Призначення**

АС «Файловий агент АРМ ВЗ» (далі – Програма), використовується для обміну файлами між АС АРМ ВЗ та іншими автоматизованими системами (АСРК, Товарообіг, тощо).

**Механізм роботи**

Програма використовує протокол FTP для обміну даними із сервером, та власний протокол обміну, побудований поверх TCP/IP, для обміну даними з утилітою віддаленого адміністрування та сервером конфігурацій. Кількість з’єднань необмежена. Кожне з’єднання ініціалізується під час роботи Програми за допомогою спеціального конфігураційного файлу, у якому зазначені всі необхідні параметри з’єднання.

Після ініціалізації з’єднання запускається окремий поток, у якому виконується процедура обміну файлами. Обмін може бути одно- або двосторонній. Після завершення процедури обміну, з’єднання переходить у режим очікування, тривалість якого вказана у відповідному параметрі конфігураційного файлу, після чого процедура обміну буде повторена, за умови, якщо не надійшла команда завершення з’єднання. Користувач також має можливість запустити процедуру обміну, не чекаючи доки завершиться час очікування.

Програма сповіщає користувача про надходження нових файлів за допомогою спливаючих повідомлень у системному треї.

Під час першого запуску Програма виводить вікно, в якому потрібно вказати індекс ВПЗ, адресу та порт сервера конфігурацій та порт для віддаленого адміністрування. Після цього Програма створює правила для брандмауеру, а також прописує шлях до виконавчого файлу у системний реєстр для автоматичного запуску після завантаження системи (якщо користувачем були зазначені вказані опції).

Ініціалізацію Програми також можна виконати через командний рядок. Для цього використовується один з наступних варіантів:

ArmAgent -init -dialog – відкриває діалогове вікно для визначення параметрів.

ArmAgent -init index=<індекс ВПЗ> server=<хост серверу конфігурацій> port=<порт серверу конфігурацій> admport=<порт адміністрування> {-firewall-add|-firewall-rem} {-auto-all|-auto-curr|-noauto} – ініціалізує Програму значеннями параметрів запуску.

ArmAgent -init -migrate <шлях до каталогу АС «Менеджер файлів АРМ ВЗ»> server=<хост серверу конфігурацій> port=<порт серверу конфігурацій> {-firewall-add|-firewall-rem} {-auto-all|-auto-curr|-noauto} – бере параметри ініціалізації з головного конфігу АС «Менеджер файлів АРМ ВЗ». При цьому процес останнього буде зупинено, а ключ на автозапуск видалено з реєстру.

ArmAgent -unreg – видаляє налаштування Програми з системного реєстру (включно із ключем на автозапуск), завершує роботу всіх запущених екземплярів Програми, а також Guardian (якщо запущений).

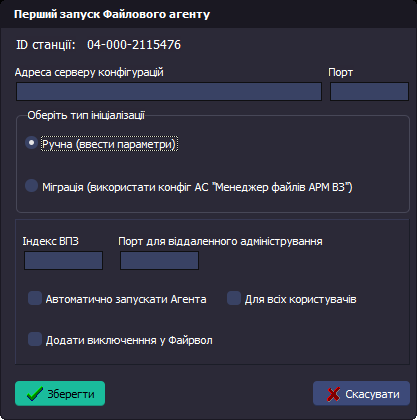
ArmAgent -firewall-add – додає виключення у брандмауер.

ArmAgent -firewall-rem – видаляє виключення з брандмауеру.

ArmAgent -auto-all – додає автозапуск для всіх користувачів (потрібні права адміністратора).

ArmAgent -auto-curr – додає автозапуск для поточного користувача.

ArmAgent -noauto – скасовує автозапуск.



Мал. 1 Приклад вікна ініціалізації Програми

**Guardian**

Guardian це окрема аплікація, яка має дві функції: контроль за безперервною роботою та перезапуск або зупинку Програми за командою. Ці функції виконує окремий поток, що слідкує за наявністю серед активних процесів процесу Програми та автоматично запускає її виконавчий файл, якщо робота Програми раптом припиняється.

Також Guardian виконує функцію оновлення Програми. Коли надходить оновлений файл виконавчого файлу Програми, вона надсилає у вікно Guardian спеціальне повідомлення. Guardian зупиняє Програму шляхом надсилання їй WinAPI-повідомлення WM\_QUIT, після чого замінює виконавчий файл. Якщо робота Програми не була коректно завершена за 10 секунд, її процес буде завершено примусово.

**Ведення логів та віддалений контроль**

Кожна дія Програми фіксується у спеціальному логу, що, задля прискорення роботи, розміщено у ОЗП комп’ютера. Запис логу у файл відбувається при завершенні роботи Програми та автоматично кожну 1 хвилину.

Роботою Програми можна керувати з іншого комп’ютера. Для цього використовується окрема утиліта, що підключається до Програми через зазначений TCP-порт. За допомогою цієї утиліти можна:

- запускати/зупиняти з’єднання із ftp-серверами.

- читати конфігурації з’єднань та Програми.

- переглядати лог роботи Програми.

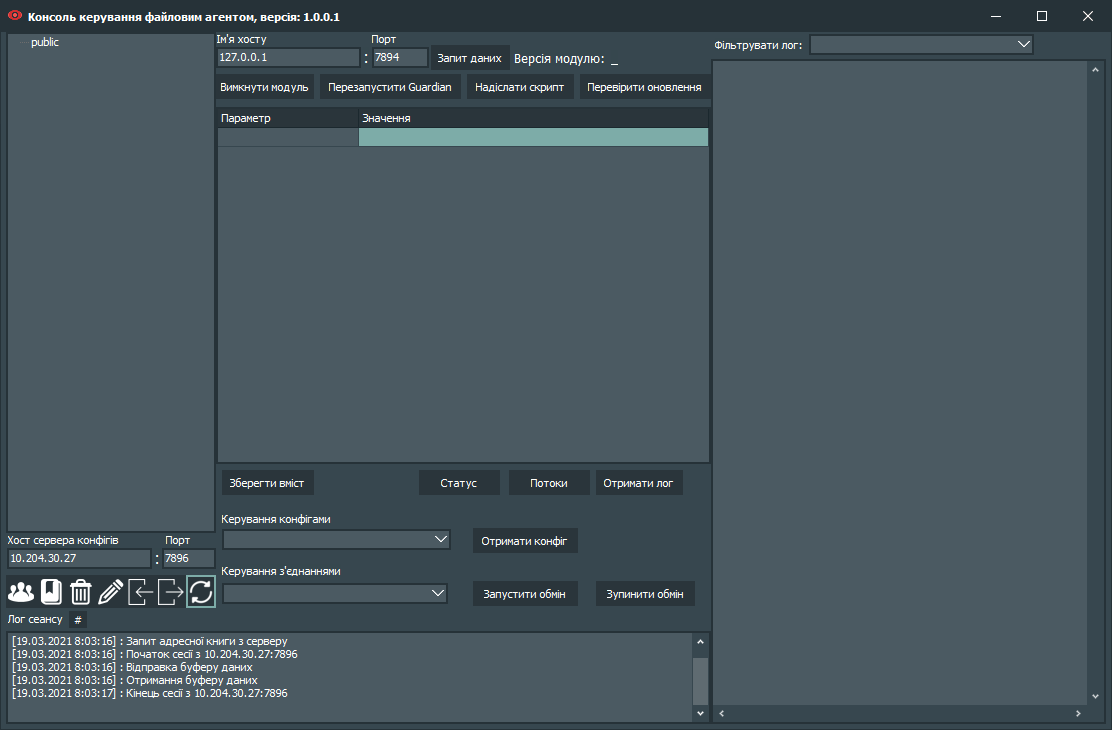
- віддалено зупиняти роботу Програми.

- надсилати Програмі керуючі скрипти.

- надсилати Програмі команду позачергової перевірки оновлень.

Консоль адміністрування має адресну книгу, в якій зберігаються всі хости, на яких запущено Програму. Користувач може додавати або видаляти записи з цієї книги, або переміщати їх між групами. Адресну книгу також можна експортувати, щоб перенести на інший комп’ютер.

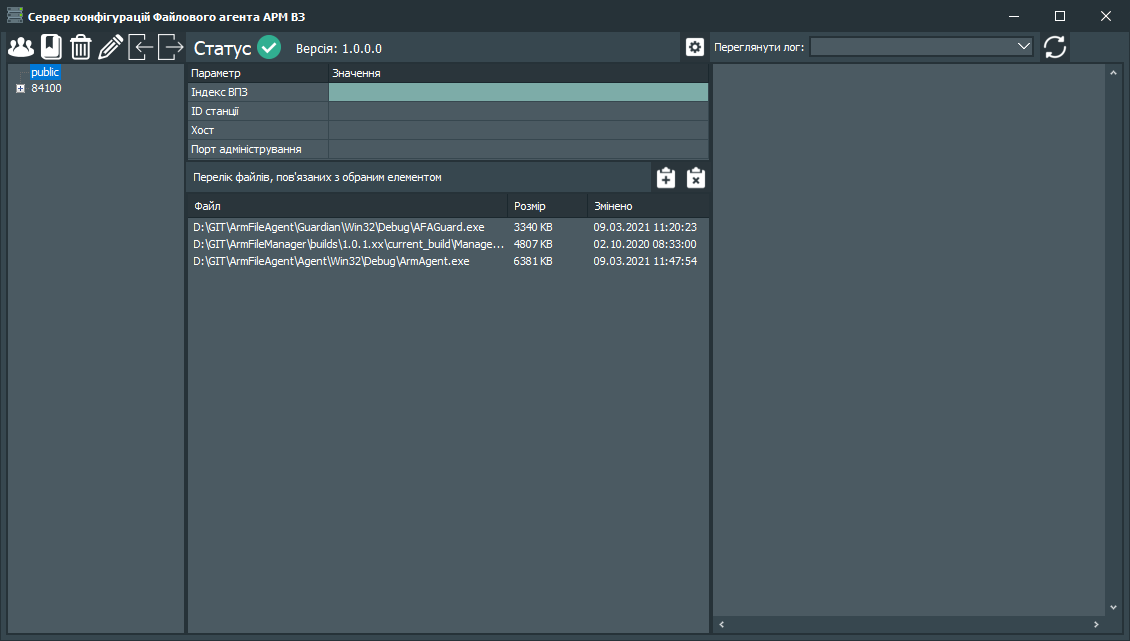
Задля зручності використання консолі адміністрування на декількох комп’ютерах, в ній передбачено режим синхронізації адресної книги із сервером конфігурацій.



Мал. 2 Зовнішній вигляд вікна консолі керування Програмою

**Сервер конфігурацій**

Сервер конфігурацій, це окрема аплікація, що зберігає перелік конфігураційних та інших файлів, що призначені для екземплярів Програми. Крім того, сервер конфігурацій веде облік підключених станцій та періодично перевіряє їх статус.



Мал. 3 Зовнішній вигляд вікна сервера конфігурацій

В лівій частині знаходиться перелік всіх підключених станцій та груп, в правій вікно перегляду логу. В центральній частині знаходиться панель інформації про обрану станцію та перелік файлів, що їй призначаються. Файли можуть бути призначені окремій станції, або групі. Записи у адресній книзі станцій створюються вручну, або автоматично, коли нова станція підключається до серверу конфігурацій. Крім того можна експортувати у файл адресну книгу станцій, або адресну книгу разом із переліком файлів, що прив’язані до кожного об’єкта книги.

**Взаємодія Програми із сервером конфігурацій**

Робочі файли Програми, а також всіх підключень, не знаходяться безпосередньо на локальному комп’ютері, де встановлена Програма. Це дозволяє максимально полегшити розгортання та перенос Програми. Всі файли зберігаються на сервері конфігурацій, що представляє собою окрему аплікацію, яка містить відповідність файлів кожному індексу ВПЗ та ідентифікатору станції.

Після запуску, Програма кожні 10 хвилин надсилає запит до серверу конфігурацій із вказанням індексу ВПЗ та ідентифікатора станції, сервер обробляє запит, знаходить відповідності та надсилає у відповідь перелік всіх файлів, які призначені для цього екземпляра Програми.

Запит до серверу від Програми має наступну структуру:

#auth%<id станції>%<індекс ВПЗ>%<порт для відправки даних>

Після обробки запиту від Програми, сервер конфігурацій формує перелік файлів, доступних для даного екземпляра Програми за наступним пріоритетом:

* Загальні файли, що призначені для всіх екземплярів Програми.
* Файли групи, що визначається індексом ВПЗ.
* Індивідуальні файли, що призначені для окремої станції.

Перелік файлів та конфігурацій надсилається у вигляді XML-документу наступної структури:

<Data type = ‘filelist’>

<File size = ’<розмір>’ version = ’<версія>’ change = ’<дата>’>ім’я файлу</File>

</Data>

Програма обробляє отриманий перелік файлів та порівнює відомості про кожний віддалений файл з наявним у локальному каталозі даних. Рішення про завантаження віддаленого файлу приймається Програмою за наступними критеріями в порядку пріоритетності:

* Локальний файл з відповідним ім’ям не існує.
* Версія віддаленого файлу більша за версію локального.
* Дата змін у віддаленого файлу більша за відповідну дату локального.
* Розмір віддаленого файлу більший за розмір локального.

Якщо жоден з цих критеріїв не задовольняється, віддалений файл не буде завантажено.

Після отримання переліку файлів для завантаження Програма по черзі запитує кожен з них з сервера.

Запит файлу має наступний вигляд:

#request%<ім’я файлу>

Після отримання файлу, виконується процес його обробки за наступними критеріями:

* Якщо це файл, він зберігається у каталог Програми, якщо це конфігурація, то вона ініціалізується одразу і розпочинає роботу.
* Якщо це виконавчий файл (.exe), перед завантаженням Програма перевіряє, чи не запущено процес цього файлу. Якщо так, Програма спробує спочатку зупинити його роботу шляхом надсилання WinAPI-повідомлення WM\_CLOSE. Якщо після цього роботу модуля не було завершено, Програма спробує примусово зупинити процес модуля. Після завантаження, модуль буде автоматично запущено.
* Якщо це командний файл (.bat, .cmd), то після завантаження його буде автоматично запущено.

Якщо в каталозі даних Програми знаходяться файли, що не включені в останній отриманий перелік файлів, вони будуть видалені з каталогу даних. При цьому якщо якісь з цих файлів були виконавчими (.exe), їх роботу буде автоматично зупинено.

**Керуючі скрипти**

Керуючі скрипти, це спеціальні файли, що містять набір інструкцій для Програми. Кожна інструкція (функція) відповідає певному функціоналу у виконавчому файлі Програми. Виконання скрипту забезпечує динамічна бібліотека ELI.dll. Кожного разу коли у Програму надходить командний скрипт (чи то з сервера конфігурацій, чи то з консолі адміністрування) Програма підключає бібліотеку ELI та транслює скрипт. Детальніше про синтаксис мови ELI можна дізнатись тут <http://eli.cc.ua> .

**Значення параметрів Програми у системному реєстрі**

**StationID** – ідентифікатор робочого місця (ім’я комп’ютера, на якому запущено Програму).

**IndexVZ** – містить індекс ВПЗ.

**RemAdmPort** – порт для підключення консолі віддаленого адміністрування.

**ConfigServerHost** – адреса серверу конфігурацій.

**ConfigServerPort** – порт для підключення до серверу конфігурацій.

**ModulePath** – шлях до виконавчого файлу Програми (потрібен для роботи Guardian).

**Значення параметрів конфігураційного файлу Програми**

**SendReportToMail** – визначає чи повинна Програма, після завершення роботи, надсилати електронного листа із логом. Приймає значення 1 чи 0.

**SmtpHost** – адреса поштового серверу для передачі повідомлень.

**SmtpPort** – порт поштового серверу.

**MailFrom** – ідентифікатор відправника електронного листа. Може містити будь-яку допустиму комбінацію символів, але рекомендується використовувати значення типу [name@mailserver.sample](mailto:name@mailserver.sample) для коректної роботи фільтрів поштового серверу. Значення не обов’язково має співпадати з реальним обліковим записом на поштовому сервері.

**MailTo** – електронна адреса, на яку будуть надіслані повідомлення.

**MailSubjectOK** – текст у полі «Тема» електронного листа, якщо оновлення було успішним.

**MailSubjectErr** – текст у полі «Тема» електронного листа, якщо сталася помилка.

**MailCodePage** – кодова сторінка для електронного листа, наприклад windows-1251, або utf-8.

**ScriptLog** – вказує, чи потрібно вести лог дій керуючого скрипту. Приймає значення 1 чи 0. Якщо встановлений у 1, інтерпретатор буде зберігати лог своїх дій у файл translate.log, розташований у робочому каталозі Програми.

**Значення параметрів конфігураційного файлу з’єднання**

**Caption** – символьний ідентифікатор підключення. Відображається у спливаючих повідомленнях та логах. Необов’язковий.

**FtpHost** – адреса ftp-серверу, на якому знаходяться файли для завантаження.

**FtpPort** – порт на якому працює ftp-сервер. Якщо не вказано, буде використаний стандартний порт 21/TCP.

**FtpUser** – ім’я користувача для авторизації на ftp-сервері.

**FtpPass** – пароль користувача для авторизації на ftp-сервері.

**MonitoringInterval –** інтервал перевірки ftp-серверу (хв.).

**TransferType** – тип передачі даних між ftp-сервером та Програмою. Приймає значення binary або ASCII.

**LeaveRemoteFiles** – визначає чи потрібно зберігати файли у каталогу на сервері після завантаження. Приймає значення 1 чи 0.

**LeaveLocalFiles** – визначає чи потрібно зберігати файли у каталогу на локальному комп’ютері після вивантаження. Приймає значення 1 чи 0.

**EnableDownload** – флаг ввімкнення режиму завантаження. Приймає значення 1 чи 0.

**RemDirDl** – каталог на сервері, з якого завантажуються файли. Якщо **EnableDownload** дорівнює 0 цей параметр ігнорується. Якщо треба вказати декілька каталогів, вони перераховуються через крапку з комою. Файли з кожного каталогу зі списку будуть завантажені у каталог, що вказаний у **LocDirDl**.

**LocDirDl** – локальний каталог у який будуть завантажені файли. Якщо **EnableDownload** дорівнює 0 цей параметр ігнорується.

**BackUpDl** – флаг ввімкнення резервного копіювання файлів, що завантажуються з серверу. Приймає значення 1 чи 0. Якщо **EnableDownload** дорівнює 0 цей параметр ігнорується.

**BackUpDirDl** – локальний каталог для резервних копій файлів, що були завантажені з серверу. Якщо **EnableDownload** дорівнює 0 цей параметр ігнорується.

**DownloadFilesMask** – перелік шаблонів імен файлів, які будуть завантажені з серверу. Використовуються типові символи-замінники. Декілька шаблонів розділюються крапкою з комою «;». Якщо **EnableDownload** дорівнює 0 цей параметр ігнорується.

**RegExDL** – визначає, чи потрібно використовувати регулярні вирази для фільтрації файлів по шаблонах. Якщо встановлений у 1, то у шаблонах з параметра **DownloadFilesMask** можна використовувати лексеми регулярних виразів. Якщо **EnableDownload** дорівнює 0 цей параметр ігнорується.

**AppendModeDL** – вмикає режим додання файлів. Якщо встановлено у 1, то з сервера будуть завантажені лише ті файли, які відсутні у локальному каталозі.

**EnableUpload** – флаг ввімкнення режиму вивантаження. Приймає значення 1 чи 0.

**RemDirUl** – каталог на сервері, у який будуть вивантажені файли. Якщо **EnableUpload** дорівнює 0 цей параметр ігнорується. Якщо треба вказати декілька каталогів, вони перераховуються через крапку з комою. Файли каталогу, що вказаний у **LocDirUl** будуть вивантажені у кожен каталог зі списку.

**LocDirUl** – локальний каталог з якого вивантажуються файли на сервер. Якщо **EnableUpload** дорівнює 0 цей параметр ігнорується.

**BackUpUl** – флаг ввімкнення резервного копіювання файлів, що вивантажуються на сервер. Приймає значення 1 чи 0. Якщо **EnableUpload** дорівнює 0 цей параметр ігнорується.

**BackUpDirUl** – локальний каталог для резервних копій файлів, що були вивантажені на сервер. Якщо **EnableUpload** дорівнює 0 цей параметр ігнорується.

**UploadFilesMask** – перелік шаблонів імен файлів, які будуть вивантажені на сервер. Використовуються типові символи-замінники. Декілька шаблонів розділюються крапкою з комою «;». Якщо **EnableUpload** дорівнює 0 цей параметр ігнорується.

**RegExUL** – визначає, чи потрібно використовувати регулярні вирази для фільтрації файлів по шаблонах. Якщо встановлений у 1, то у шаблонах з параметра **UploadFilesMask** можна використовувати лексеми регулярних виразів. Якщо **EnableUpload** дорівнює 0 цей параметр ігнорується.

**AppendModeUL** – вмикає режим додання файлів. Якщо встановлено у 1, то на сервер будуть вивантажені лише ті файли, які відсутні у віддаленому каталозі.

**RunOnce** – змушує підключення виконати процедуру обміну лише один раз, після чого зупинитись. Приймає значення 1 чи 0.

**StartAtTime** – вказує, чи повинно підключення виконати обмін в певний час. Приймає значення hh:mm. Якщо не встановлений, то ігнорується. Якщо на момент ініціалізації з’єднання системний час перевищує значення параметру, процедура обміну буде запущена негайно.

**BackUpKeepDays** – кількість днів, впродовж яких бекапи оброблених файлів будуть зберігатися на диску.

**DownloadFromSubDirs** – вмикає пошук файлів також у підкаталогах каталогу, що вказаний у параметрах **RemDirDl** або **LocDirUl**. Глибина вкладеності – один рівень. Приймає значення 1 чи 0.

**SaveWithSubDirs** – змушує Програму зберігати файли, дотримуючись структури підкаталогів під час завантаження та вивантаження. Глибина вкладеності – один рівень. Приймає значення 1 чи 0.

**Перелік змінних що використовуються у конфігураційних фалах підключень**

**$IndexVZ** – значення параметру **IndexVZ** з системного реєстру Програми.

**$StationID** – значення параметру **StationID** з системного реєстру Програми.

**$Date** – поточна системна дата.

**$AppPath** – повний шлях до каталогу, з якого запущено Програму.

**$DataPath** – повний шлях до робочого каталогу Програми.