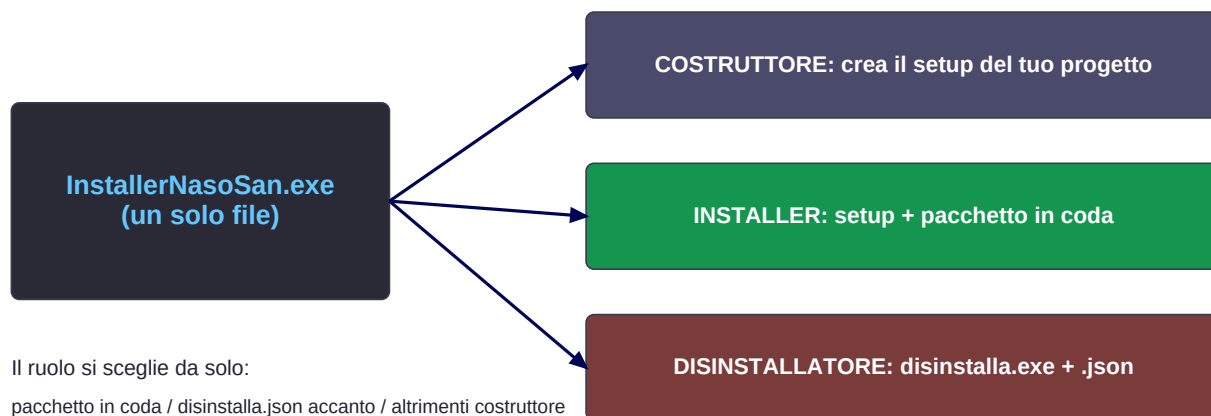


Guida programmatore

InstallerNasoSan e' un generatore di installer per Windows (C#, .NET Framework 4.8, un solo exe). Sostituisce Inno Setup: da un progetto gia' compilato produce un solo file di setup, senza componenti da installare prima. Questa guida spiega come configurarlo e integrarlo.

1. Schema generale



Il pacchetto e' accodato dopo l'ultima sezione del PE: una firma digitale aggiunta dopo non lo rovina.

Il setup generato e' una copia dell'exe dello strumento con in coda un pacchetto: SHA-256 verificato prima di scrivere qualsiasi file, zip con i file **alla radice** (nessuna cartella contenitore). Non serve niente di installato in piu' sul computer di destinazione (solo .NET Framework 4.8, presente in Windows 10/11).

2. Uso rapido

Passo	Azione
1	Compila il tuo programma: gli exe devono stare in compilato\ (o indicali con "origine").
2	Crea installer.json nella cartella del progetto (esempi in esempi\) oppure usa la finestra guidata.
3	Costruisci: InstallerNasoSan.exe /costruisci /progetto=<cartella> oppure lancia compila_progetto.ps1 -exe.
4	Il setup compare in compilato\ (e in P:\ con compila_progetto). Firmalo se vuoi: il pacchetto non si rovina.

```
{
  "nome": "MioProgramma",
  "editore": "Mio Nome",
  "exe": [ { "file": "MioProgramma.exe", "nome": "Mio Programma" } ]
}
```

La versione dell'applicazione **non si scrive** in installer.json: si legge da app.json (campo "versione"). Il nome del file di setup segue questa priorita': /nome, poi nome artefatto di app.json, poi il nome del progetto.

3. installer.json campo per campo

Tutti i campi sono facoltativi salvo "exe". Un campo assente prende il valore delle impostazioni generali (cfg\cfg_generale_InstallerNasoSan.json, modificabili dall'ingranaggio), poi dei valori predefiniti.

Campo	Tipo / valori	Significato
nome	testo	Nome dell'app. Se vuoto: nome di app.json, poi nome progetto. Nel titolo compare senza la sigla "NasoSan" (logica 003).
nomevisualizzato	testo	Nome mostrato nelle finestre e in "App e funzionalita".
identificativo	testo	Chiave interna (istanza singola, registro). Predefinito dal nome, sanificato.
versione	testo	Sconsigliato: si legge da app.json.
editore, sito, descrizione	testo	Mostrati nella pagina di benvenuto e in "App e funzionalita".
sottocartella	testo	Cartella dell'app sotto la cartella base. Predefinito: nome.
nomesetup, cartellaoutput	testo	Nome del file di setup (".exe" aggiunto) e cartella di uscita (predefinita: compilato).
exe[]	lista di oggetti	Programmi installati insieme (obbligatorio, almeno uno). Vedi tabella sotto.
contenutopredefinito	vero/falso	Include automaticamente exe, cfg, media e app.json (predefinito vero).
includi[]	testo oppure {da,a}	File o cartelle extra. "da" puo' contenere *; "a" e' la destinazione relativa. Cartella intera ammessa.
escludi[]	modelli	Esclusioni. Senza "/" vale sul nome in ogni cartella; con "/" sul percorso intero; * anche tra cartelle, ? un carattere.
nonsovrascrivere[]	modelli	File che negli aggiornamenti non si sovrascrivono e non si rimuovono (predefinito cfg/*, log/*).
processidachiudere[]	nomi	Altri processi da chiudere prima di installare (oltre agli exe dell'elenco).
modoinstallazione	scelta utente tutti	Vincola l'ambito. "scelta" lascia decidere l'utente.
modopredefinito	utente tutti	Ambito preselezionato nel wizard.
chiusuraprogrammi	{modo, attese secondi, riapri}	modo: automatica chiedi solocortese. Attesa 1-120 s.
migrazione	chiedi automatica mai	Come trattare un'installazione in un'altra cartella.
avviaautomatico	{modo}	chiave (Run, predefinito) oppure task (attivita' pianificata).
idinnoprecedente	testo	AppId Inno Setup della vecchia versione (rileva la vecchia installazione).
posizioniprecedenti[]	percorsi	Cartelle di vecchie installazioni da rilevare e migrare (variabili d'ambiente ammesse).
firewall[]	{nome, protocollo, porta, exe}	Regola in ingresso (netsh), solo con amministratore (logica 106).
scorciatoiadisinstalla	vero/falso	Collegamento "Disinstalla" nel menu Start.
licenza, logo, icona	percorsi	Testo licenza (pagina con accettazione), logo PNG (max 1 MB), icona .ico del setup.
logattivo	vero/falso	Registro attivita' acceso.

Campi di ogni elemento di exe[]

Campo	Predefinito	Significato
file	(obbligatorio)	Percorso dell'exe dentro la cartella di installazione.

Campo	Predefinito	Significato
origine	cerca nel progetto, poi in compilato\	Dove prenderlo al momento della costruzione.
nome	nome del file	Nome nei collegamenti (spazi e accenti conservati).
scorciatoiamenu	vero	Collegamento nel menu Start.
scorciatoiadesktop	falso	Collegamento sul desktop.
avvioautomatico	falso	Avvio con Windows.
avviadopo	vero solo per il primo	Avvia al termine (senza privilegi anche se il setup e' elevato).
parametri, cartellalavoro	vuoti	Argomenti e cartella di lavoro dei collegamenti.

4. Riga di comando e codici di uscita

Ruolo	Opzioni
Costruttore	/costruisci /progetto= /config= /uscita= /nome= /cfg= /stub= /risultato= /log
Setup	/silenzioso /utente /tutti /cartella= /chiusura= /migrazione= /senzaavvio /nomenu /desktop /avvioautomatico /log
Disinstallatore	/silenzioso /rimuovidati
Comune	/aiuto (senza argomenti il costruttore apre la finestra guidata)

Codice	Significato
0	Riuscito
1	Configurazione o argomenti non validi (messaggio in italiano)
2	Annullato dall'utente
3	Errore file o pacchetto (danneggiato, troncato, scrittura fallita)
4	Servono i permessi di amministratore
5	Programmi non chiusi
6	Gia' in esecuzione
9	Errore imprevisto

Con `/risultato=file.json` il costruttore scrive
`{"ok":true,"setup":"percorso","errore":""}`: e' quello che legge `compila_progetto`.

5. Integrazione con `compila_progetto.ps1`

Il ramo "exe" resta com'era per i progetti con `.iss` (ISCC) e per il solo `csproj` (publish self-contained). Novita' additive: se nella cartella del progetto esiste **installer.json** (e non c'e' un `.iss`) il setup lo costruisce `InstallerNasoSan`; se la consegna ha **compilazione="dotnetbuild"** viene eseguito prima `dotnet build -c Release` del `csproj`. Senza `installer.json` l'exe compilato e' l'artefatto stesso (copiato in `compilato\` e in `P:\`).

```
{ "nome": "MioProgramma", "versione": "1.20",
  "tipoconsegna": "exe", "compilazione": "dotnetbuild",
  "percorsosorgente": "MioProgramma.csproj",
  "nomeartefatto": "MioProgramma_setup.exe" }
```

Percorso dello strumento: campo `consegna` `strumentoinstaller`, poi variabile d'ambiente `NASOSAN_INSTALLER`, poi `script\InstallerNasoSan\compilato\InstallerNasoSan.exe`. Se non lo trova, la consegna termina con errore e messaggio chiaro.

6. Compilare i sorgenti di `InstallerNasoSan`

```
cd script\InstallerNasoSan
dotnet build InstallerNasoSan.csproj -c Release
(uscita: compilato\InstallerNasoSan.exe)
```

Struttura: file base `InstallerNasoSan.cs` (solo Main) e 17 file-sezione `InstallerNasoSanN.cs`; la logica numerata e' in `logica_InstallerNasoSan.md` (un punto di primo livello = un file). La versione sta solo in `app.json`. Il logo `NasoSan` si copia in `media\icon.png` (senza, l'intestazione mostra solo il nome).

7. Cosa fa il motore (in breve)

Funzione	Comportamento
Aggiornamento sul posto	Se esiste un'installazione di questo installer, cartella e ambito predefiniti sono i suoi <code>cfg\</code> e <code>log\</code> non si sovrascrivono.
Migrazione	Da installer proprio, Inno (idinnoprecedente) o cartelle note: copia completa, eliminazione della vecchia solo a fine lavoro riuscito. Cartelle annidate: niente migrazione.

Funzione	Comportamento
Scrittura sicura	File temporaneo poi sostituzione, ritentativi se bloccato; protezione zip-slip (nessun ".." ne' percorso assoluto).
Chiusura programmi	Per cartella, mai solo per nome se il percorso e' leggibile; se restano attivi non scrive nulla.
Tracciabilita'	disinstalla.json elenca file, collegamenti, avvii, regole firewall e chiave di registro creati.
App e funzionalita'	Voce con disinstallazione normale e silenziosa, dimensione e data.

8. Cosa e' stato provato e cosa no

*Provati automaticamente con mono e display virtuale: costruzione, pacchetto, installazione, aggiornamento, migrazione, chiusura programmi, disinstallazione, pacchetto ostile, log, finestre (127 controlli sul nucleo, 27 sulle finestre). **Da provare su Windows reale:** aspetto e DPI, titlebar scura, collegamenti (WScript.Shell), registro, UAC, attivita' pianificata, regole firewall, icona e informazioni versione dell'exe di setup. Non gestito: riavvio per sostituire file bloccati (dopo i ritentativi l'installazione si ferma con errore 3).*

9. Licenza

Creative Commons Attribution 4.0 (CC BY 4.0) - nasosan.it -

<https://creativecommons.org/licenses/by/4.0/>. Puoi copiare, modificare e usare anche a scopo commerciale, mantenendo l'attribuzione (file NOTICE). Il piede firma dei setup generati cita l'attribuzione. I programmi impacchettati restano dei rispettivi autori.