

Peregrine

ServiceCenter

Guida all'installazione e alla configurazione del client Java

Versione 5.1

Copyright © 2002-2003 Peregrine Systems, Inc. o sue consociate. Tutti i diritti riservati.

Le informazioni contenute in questo documento sono di proprietà di Peregrine Systems, Incorporated e possono essere utilizzate o diffuse solo previa autorizzazione scritta di Peregrine Systems, Inc. Non è consentito riprodurre questo documento, in tutto o in parte, se non previa autorizzazione scritta di Peregrine Systems, Inc. In questo documento sono citati i nomi commerciali di numerosi prodotti. Nella maggior parte dei casi, se non in tutti, tali designazioni sono indicate come marchi o marchi registrati delle rispettive società.

Peregrine Systems® e ServiceCenter® sono marchi registrati di Peregrine Systems, Inc. o delle sue consociate.

Questo documento ed il relativo software descritto nella guida vengono forniti con contratto di licenza o di non divulgazione e possono essere utilizzati o copiati solo in conformità con i termini di tale contratto. Le informazioni contenute in questo documento sono soggette a modifiche senza preavviso e non rappresentano un impegno da parte di Peregrine Systems, Inc. Contattare l'Assistenza Clienti di Peregrine Systems, Inc. per verificare la data della versione più recente di questo documento.

I nomi delle aziende e delle persone utilizzati nel database esemplificativo e negli esempi dei manuali sono fittizi e citati esclusivamente per illustrare l'uso del software. Qualsiasi somiglianza con nomi di aziende o persone reali, del passato o del presente, è puramente casuale.

Per ottenere assistenza tecnica su questo prodotto o per richiedere documentazione relativa a un prodotto per il quale si dispone di licenza, inviare un messaggio e-mail all'Assistenza Clienti di Peregrine Systems, Inc. all'indirizzo support@peregrine.com.

Se si desidera sottoporre commenti o suggerimenti su questa documentazione, inviare un messaggio e-mail all'ufficio Pubblicazioni Tecniche di Peregrine Systems, Inc. all'indirizzo doc_comments@peregrine.com.

Questa edizione si riferisce alla versione 5.1 del programma concesso in licenza.

Peregrine Systems, Inc.
3611 Valley Centre Drive San Diego, CA 92130
Tel. +1 800 638 5231 o +1 858 481 5000
Fax +1 858 481 1751
www.peregrine.com



Sommario

	Guida introduttiva.	7
	Conoscenze richieste.	8
	Esempi	9
	Come contattare l'Assistenza Clienti	9
	Sito Web CenterPoint di Peregrine	9
	Sede centrale.	9
	America del Nord e America del Sud.	10
	Europa, Asia/Pacifico, Africa	10
	Come contattare Education Services	10
Capitolo 1	Informazioni preliminari	11
	Altre funzioni	12
	Vantaggi e limiti del client Java	13
	Limitazioni	14
	Funzioni di installazione.	14
	Requisiti.	15
	Considerazioni sull'installazione.	15
	Client basati su browser	15
	Client autonomi	17
	Supporto plug-in	18

Capitolo 2	Installazione e avvio del Client Java	19
	Installazione di un client Java autonomo	20
	Esecuzione di un client autonomo	31
	Aggiornamento di un client autonomo	31
	Installazione di un client Java basato su browser	32
	Esecuzione di un client Java basato su browser	35
	Modifica delle dimensioni heap del client Java.	39
	Sistemi operativi Unix	40
	Esecuzione di un client autonomo	42
	Esecuzione di un client basato su browser.	44
	Modifica delle dimensioni heap del client Java.	45
	Sistemi operativi Macintosh	46
	Installazione di MRJ per OS 9.x	46
	Installazione del client Java: OS X	51
	Esecuzione di un client autonomo	53
	Esecuzione di un client basato su browser.	53
	Sistemi operativi OS/2	53
	Esecuzione di un client autonomo	54
	Installazione del Java Runtime Environment	54
	Velocità di connessione	55
	La Console Java	56
Capitolo 3	Configurazione del Client	59
	Impostazione delle preferenze nei file .htm	62
	Parametri Dictionarydir e Downloaddictionarydir	63
	Impostazione delle preferenze in scj.ini	63
	Impostazione delle preferenze in scjpref.ini	65
	Impostazione delle preferenze in sc.ini	66
	Parametro explorer	68
	Parametro explorerdefault	69
	Esempio: clientprinting	70
	Esempio: explorerhome	70
	Impostazione delle preferenze dalla riga di comando	75
	Aggiornamento automatico degli elenchi di record	76
	Codici delle lingue.	77
	Esempio: Supporto per la lingua turca dell'applicazione autonoma	78

Capitolo 4	Server Hub	79
	Utilizzo di servlet Java	80
	Requisiti per il supporto di servlet	81
	Parametri del Server Hub	81
	Scenari di installazione.	83
	Connessione callback	84
	Passaggio 1: Configurazione del motore servlet	85
	Passaggio 2: Configurazione del client Java	90
	Connessione diretta	93
	Passaggio 1: Configurazione del motore servlet	94
	Passaggio 2: Configurazione del client Java	95
	Connessione diretta senza server HTTP.	96
	Passaggio 1: Configurazione del motore servlet	97
	Passaggio 2: Configurazione del client Java	97
	Configurazioni con firewall.	99
	Client Java protetto da firewall	99
	Server hub protetto da firewall	100
	Supporto SSL del client Java	102
	Requisiti di sistema per SSL.	102
	Abilitazione del supporto SSL.	102
	Supporto SSL lato server	103
	Recupero delle librerie SSL (solo JRE 1.3.x)	105
	Supporto SSL lato client	105
	Verifica SSL per server e client	109
	Pacchetti di crittografia predefiniti.	109
Capitolo 5	ServiceInfo Universal	111
	Utilizzo di servlet Java	112
	Parametri SIU	113
	Parametri del file HTML	114
Capitolo 6	Risoluzione dei problemi	117
	Sistemi Unix	120
	Sistemi Macintosh.	121
	Server Hub.	122
	Tutti i sistemi.	123

Capitolo 7	Specifiche di accessibilità	125
	Funzioni da tastiera	126
	Preferenze di visualizzazione	127
	Ausili tecnologici	127
	Strumenti per utenti offerti da terze parti.	127
	Strumenti di sviluppo	128
	Impostazione delle opzioni di modifica.	129
	Impostazione delle opzioni di visualizzazione	129
	Aspetti relativi alla conformità alla Section 508.	130
	Funzioni non disponibili.	131
Indice		133

Guida introduttiva

Il client Java di Peregrine Systems, Inc. è un'applicazione che consente di visualizzare ServiceCenter con un browser Web o come applicazione autonoma utilizzando un Java Runtime Environment (JRE) locale. Questa guida fornisce istruzioni per l'installazione e la configurazione del client Java.

La *Guida all'installazione e alla configurazione del client Java* contiene le informazioni riportate di seguito:

- *Guida introduttiva* descrive la presente guida e le conoscenze necessarie per gli utenti. Fornisce informazioni di assistenza per il prodotto, elenca i requisiti di sistema per le piattaforme client per il client/server Windows di ServiceCenter e indica come contattare l'assistenza clienti di Peregrine Systems, Inc.
- *Informazioni preliminari* a pagina 11 fornisce una breve introduzione al client Java, comprendente alcuni punti da tenere in considerazione prima dell'installazione e la descrizione del funzionamento del programma di installazione del client Java.
- *Installazione e avvio del Client Java* a pagina 19 fornisce le istruzioni per l'installazione e l'avvio del client Java in un browser Web o come applicazione autonoma su tutte le piattaforme.
- *Configurazione del Client* a pagina 59 descrive le modalità di configurazione dei parametri e delle preferenze utente per i file `scjava.launch.htm` e `sc.ini`.

- *Server Hub* a pagina 79 descrive le modalità di installazione e configurazione del server hub.
- *ServiceInfo Universal* a pagina 111 fornisce istruzioni per il collegamento del sistema al ServiceInfo Universal (SIU) e la relativa configurazione.
- *Risoluzione dei problemi* a pagina 117 fornisce informazioni per la risoluzione dei problemi relativi alla configurazione autonoma e con applet.
- *Specifiche di accessibilità* a pagina 125 descrive le nuove caratteristiche del client Java relative a facilità di utilizzo, funzionalità e integrazione con software per disabili sviluppato da terze parti al fine di garantire la conformità alla Section 508 del Rehabilitation Act statunitense.

Conoscenze richieste

Le istruzioni contenute nella guida prevedono la conoscenza operativa di ServiceCenter e della piattaforma di installazione. Ulteriori informazioni sono disponibili nelle guide elencate di seguito.

- Per informazioni su una determinata piattaforma, consultare la documentazione relativa alla piattaforma specifica.
- Per ulteriori informazioni su come personalizzare i parametri per l'ambiente di lavoro, vedere la guida *ServiceCenter Technical Reference* (in inglese).
- Prima di avviare il server di ServiceCenter, consultare la *Guida per l'utente di ServiceCenter*.
- Le informazioni sull'amministrazione e sulla configurazione del sistema sono reperibili nella *Guida per l'amministratore del sistema di ServiceCenter* e nella *Guida per l'amministratore delle applicazioni di ServiceCenter*.
- La configurazione del database è descritta nella *ServiceCenter Database Management and Administration Guide* (in inglese).
- Le copie delle guide in formato PDF possono essere scaricate dal sito Web CenterPoint utilizzando Adobe Acrobat Reader, disponibile sul sito. Per ulteriori informazioni, vedere *Sito Web CenterPoint di Peregrine* a pagina 9. È inoltre possibile ordinare copie cartacee della documentazione contattando il proprio rappresentante Peregrine Systems.

Esempi

Le finestre e gli esempi riportati in questa guida hanno scopo meramente illustrativo e potrebbero differire dai casi specifici.

Come contattare l'Assistenza Clienti

Per maggiori informazioni e assistenza in merito alla nuova versione o a ServiceCenter in generale, contattare l'Assistenza Clienti di Peregrine Systems.

Sito Web CenterPoint di Peregrine

Informazioni sulla compatibilità tra versioni, sui requisiti hardware e software e su altri aspetti della configurazione sono disponibili presso il sito Web CenterPoint di Peregrine all'indirizzo <http://support.peregrine.com>

- 1 Immettere il nome e la password di accesso.
- 2 Selezionare **Go** per passare a **CenterPoint**.
- 3 Per informazioni sulla configurazione e sulla compatibilità scegliere **ServiceCenter** dall'elenco **My Products** nella parte superiore della pagina.

Nota: Per informazioni sulle sedi di assistenza locali selezionare **Whom Do I Call?** dalla sezione **Contents** sulla sinistra della pagina. Verrà visualizzata la pagina **Peregrine Worldwide Contact Information**.

Sede centrale

Indirizzo: Peregrine Systems, Inc.
Attn: Customer Support
3611 Valley Centre Drive
San Diego, CA 92130

Telefono: +1 (858) 794-7428

Fax: +1 (858) 480-3928

America del Nord e America del Sud

Telefono: +1 (800) 960-9998 (solo Stati Uniti e Canada, numero verde)
+ 1 (858) 794-7428 (Messico, America centrale e
America del Sud)

Fax: +1 (858) 480-3928

Posta elettronica: support@peregrine.com

Europa, Asia/Pacifico, Africa

Per informazioni sulle sedi locali, vedere *Sito Web CenterPoint di Peregrine*.
È possibile anche contattare la *Sede centrale*.

Come contattare Education Services

Vengono offerti servizi di formazione per l'intera gamma di prodotti
Peregrine, compreso ServiceCenter.

Informazioni aggiornate sui servizi di formazione offerti possono
essere ottenute presso la sede indicata di seguito o all'indirizzo:

<http://www.peregrine.com/education>

Indirizzo: Peregrine Systems, Inc.
Attn: Education Services
3611 Valley Centre Drive
San Diego, CA 92130 USA

Telefono: +1 (858) 794-5009

Fax: +1 (858) 480-3928

1 Informazioni preliminari

CAPITOLO

Il client Java di ServiceCenter costituisce un'interfaccia Java per le applicazioni di ServiceCenter. Il client Java supporta sia le funzionalità del client Windows, sia le funzionalità avanzate specifiche del client Java e dell'ambiente Java.

Leggere questo capitolo per ottenere informazioni su

- *Caratteristiche del Client Java* a pagina 12
- *Vantaggi e limiti del client Java* a pagina 13
- *Requisiti* a pagina 15
- *Considerazioni sull'installazione* a pagina 15
- *Supporto plug-in* a pagina 18

Caratteristiche del Client Java

Il client Java offre varie funzionalità che lo rendono più interessante del client Windows in alcune situazioni. Il client Java fornisce quanto segue.

- Un riquadro di spostamento con struttura ad albero denominato ServiceCenter Explorer
- Supporto Multiple Document Interface (MDI)
- Una barra degli strumenti Preferiti
- Accesso ai moduli usati di frequente
- Opzione Ripristina moduli all'avvio
- Funzionalità di sessioni multiple
- Indicatore attività nella barra di stato
- Pulsante messaggi nella barra di stato
- Salvataggio Windows all'uscita
- Collegamenti ipertestuali in un campo di testo

Altre funzioni

Trattandosi di un prodotto Java, il client può essere eseguito ovunque sia installato un ambiente Java. Il client può essere configurato per l'esecuzione come applicazione autonoma oppure da un server Web, in cui il client viene scaricato in remoto ed eseguito in un browser Web. L'ubicazione del server Web e la relazione client/server Java viene specificata dall'amministratore di ServiceCenter durante l'installazione del client Java oppure configurando i file HTML che avviano l'esecuzione del client nel browser.

Il client Java supporta Input Method Editor (IME) di Microsoft. IME è uno strumento Microsoft che consente la digitazione di testo in lingue con un gran numero di caratteri, come il giapponese.

Il client Java utilizza socket TCP/IP per comunicare con un server di ServiceCenter standard. È possibile installare e avviare il client Java senza dover apportare alcuna modifica all'ambiente del server esistente.

Vantaggi e limiti del client Java

L'utilizzo del client Java comporta diversi vantaggi. Non è richiesta alcuna amministrazione sul lato client. Tutti i client Java collegati possono eseguire il download di aggiornamenti non appena questi sono resi disponibili sul server dall'amministratore. È possibile aggiornare il client Java alla versione più recente senza dover aggiornare il server o le applicazioni di ServiceCenter. Ad esempio:

- Il client Java 3.0 SP3 è compatibile con tutti i server versione 3.0.
- Il client Java 4.0 è compatibile con tutti i server versione 3.0 e 4.0.
- Il client Java 5.x è compatibile con tutti i server versione 3.0 e 4.0.

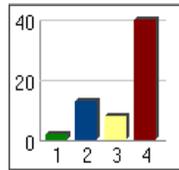
Tutte le applicazioni di ServiceCenter sono supportate senza necessità di interventi di modifica o personalizzazione. L'interfaccia del client Java può avere lo stesso stile grafico del client Windows oppure, per gli spostamenti all'interno del client, si può utilizzare ServiceCenter Explorer (la struttura ad albero).

Dal client Java è possibile stampare, allegare file oppure utilizzare il supporto OLE (Object Linking and Embedding) per creare oggetti e collegarli o incorporarli nel client Java. Dato che il numero di strumenti e librerie che supportano Java cresce rapidamente, è del tutto probabile che il client Java possa integrarsi perfettamente in un ambiente Java anche molto sofisticato.

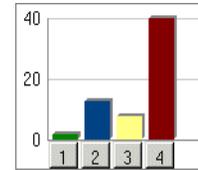
Limitazioni

Il client Java presenta alcune limitazioni:

- I pulsanti dei grafici vengono visualizzati nel client Windows, ma non nel client Java. Sono disponibili le stesse funzionalità, tuttavia è necessario fare clic sulla barra grafica del client Java e quindi scegliere i pulsanti nel client Windows.



Client Java, senza pulsanti



Client Windows, con pulsanti

- Poiché Java è in continua evoluzione, non è disponibile lo stesso livello di supporto su tutte le piattaforme.

Il client Java ServiceCenter si basa sulle classi Java contenute nelle API Java standard, compreso l'Abstract Window Toolkit (AWT) di Sun. AWT fa parte delle Java Foundation Classes (JFC), che rappresentano l'API standard per le interfacce grafiche utente Java. Tutte le classi necessarie per il client Java di ServiceCenter, compresi i componenti Swing di JFC, sono incluse nell'installazione di questa versione.

Funzioni di installazione

Il programma di installazione Java consente di evitare di ripetere più volte il download dei file del client e delle librerie, riducendo l'occupazione di banda e avviando il client più rapidamente. Il programma di installazione rileva automaticamente gli aggiornamenti del client eventualmente presenti sul server e, all'esecuzione successiva, notifica all'utente la possibilità di aggiornamento. Rileva i problemi nell'ambiente dell'utente prima dell'avvio del client.

Il programma di installazione consente di eliminare facilmente i file relativi al client Java. All'eliminazione del client Java, vengono eliminati anche tutti i file correlati. Le cartelle di installazione non verranno eliminate se contengono file creati dall'utente.

Requisiti

Durante l'installazione è necessario specificare il nome dell'host TCP/IP e l'indirizzo del servizio del server di ServiceCenter. In questa configurazione il client Java deve instaurare una connessione diretta TCP/IP al server di ServiceCenter o collegarsi mediante un server hub.

Considerazioni sull'installazione

Il client Java di ServiceCenter può essere installato e configurato come client autonomo o basato su browser. Prima di eseguire il programma di installazione, esaminare le opzioni offerte dai due tipi di installazione e scegliere quella che risponde meglio alle proprie esigenze.

Client basati su browser

Il client Java basato su browser può essere installato sullo stesso computer del server Web oppure per un collegamento remoto con il server Web. La Figura 1-1 a pagina 16 mostra la relazione tra un server di ServiceCenter e vari client Java.

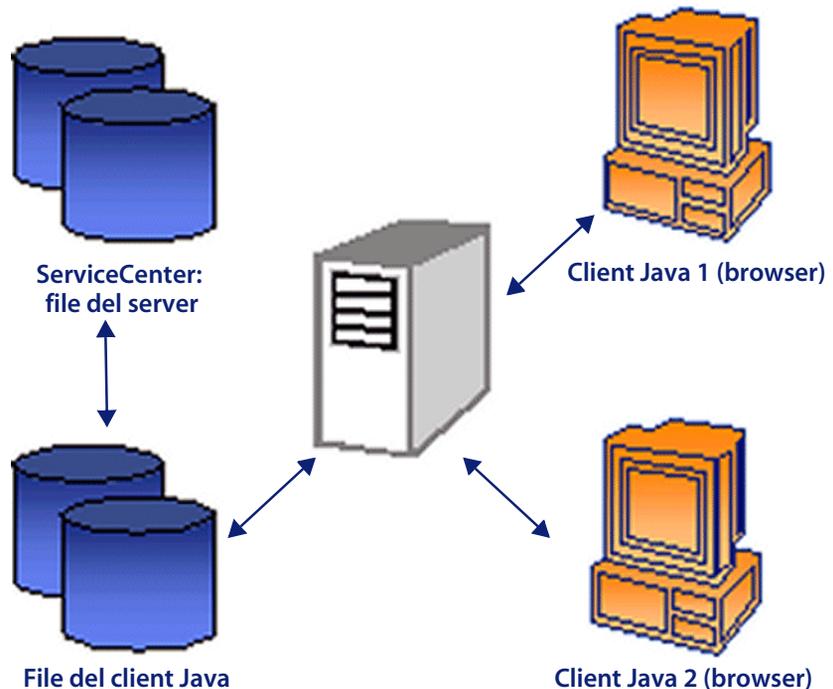


Figura 1-1: Relazione client/server Java

Il client Java basato su browser può essere eseguito da Internet Explorer e Netscape Navigator. Per consentire l'accesso al server mediante il browser, il prodotto deve essere installato su un server Web o un'unità di rete. Per avviare il client basato su browser, gli utenti del client Java devono specificare l'URL del file `scjava.launch.htm`. Per informazioni sulla compatibilità con le varie versioni dei browser, vedere *Sito Web CenterPoint di Peregrine* a pagina 9.

File di avvio

Il file `scjava.launch.htm` che avvia l'esecuzione del client Java nasconde all'utente i comandi del browser (gli strumenti e le barre dei menu normali del browser), visualizzando solo i comandi di ServiceCenter. Il file `scjava.htm` che avvia l'esecuzione del client Java visualizza sia i comandi del browser che quelli di ServiceCenter. Gli utenti Macintosh possono utilizzare il file `scjavamac.htm` per visualizzare sia i comandi del browser che i comandi di ServiceCenter.

Alcuni sistemi Unix non sono in grado di gestire il meccanismo di firma utilizzato dal client Java. Gli utenti del client Java su tali sistemi possono collegarsi mediante il file `scapplelet.htm`. È possibile utilizzare questo metodo di connessione anche nei casi in cui non si dispone di autorizzazione di scrittura sulle directory del browser.

Client autonomi

Se è stato installato un JRE supportato, il client Java può essere eseguito anche come applicazione autonoma. La Figura 1-2 mostra la relazione locale tra un client Java autonomo e il server di ServiceCenter.

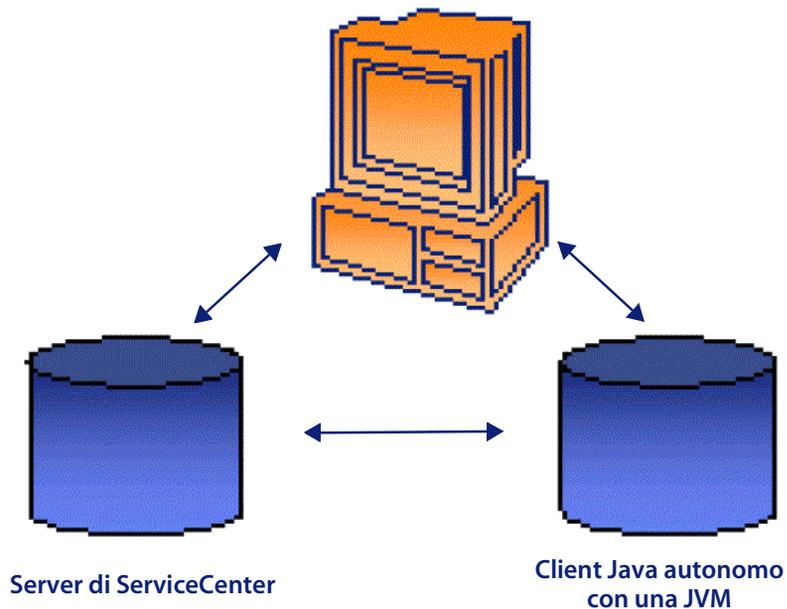


Figura 1-2: Relazione client Java autonomo/server

Il client Java può essere installato anche come applicazione autonoma eseguibile grazie alla Java Virtual Machine (JVM). Ad esempio, vi sono varie versioni del Java Runtime Environment (JRE) di Sun che creano un ambiente JVM compatibile. Anche altri JRE, ad esempio Microsoft JView, sono in grado di supportare il client Java. Per informazioni sulla compatibilità, vedere *Sito Web CenterPoint di Peregrine* a pagina 9.

Per creare client Java autonomi è necessario eseguire il programma di installazione su ciascun computer client. Durante l'installazione è necessario specificare il nome dell'host TCP/IP e l'indirizzo del servizio del server di ServiceCenter. Nel caso in cui sia il server che il client autonomo risiedano sulla stessa workstation o server, il nome host TCP/IP e l'indirizzo del servizio indicheranno tale computer.

Supporto plug-in

Il client Java può sfruttare la tecnologia Java Plug-in di Sun. È possibile eseguire il download e installare il Java Plug-in dal sito Web di Sun per eseguire il client Java (o qualsiasi altra applet) utilizzando il JRE di Sun più recente. Il client Java di ServiceCenter è compatibile con JRE meno recenti. Netscape versione 6.0 e successive includono il supporto per il Plug-in Java.

Windows XP non comprende il supporto Java, anche se alcuni produttori e rivenditori di hardware inseriscono nei propri sistemi una vecchia versione del JRE. L'amministratore del sistema può verificare se sul sistema è installato il supporto Java. Il JRE può essere scaricato dal sito Web di Sun. Per ulteriori informazioni, vedere *Installazione del Java Runtime Environment* a pagina 54.

2 Installazione e avvio del Client Java

CAPITOLO

Questo capitolo descrive l'installazione del client Java su diverse piattaforme, per l'esecuzione come applet in un browser e come applicazione autonoma. Per ulteriori informazioni sull'installazione di ServiceCenter su un sistema operativo Windows, vedere la *Guida all'installazione client/server per Windows*. Per ulteriori informazioni sull'installazione di ServiceCenter su un sistema operativo Unix, vedere la *Guida all'installazione client/server per Unix*.

Leggere queste sezioni per ulteriori informazioni sull'installazione del client Java.

- *Sistemi operativi Windows* a pagina 20
- *Sistemi operativi Unix* a pagina 40
- *Sistemi operativi Macintosh* a pagina 46
- *Sistemi operativi OS/2* a pagina 53
- *Installazione del Java Runtime Environment* a pagina 54
- *Velocità di connessione* a pagina 55
- *La Console Java* a pagina 56

Sistemi operativi Windows

Il client Java può essere eseguito come client autonomo, come client basato su browser locale su un server Web oppure come client basato su browser remoto.

Installazione di un client Java autonomo

Un client Java autonomo può essere installato seguendo la procedura illustrata in questa sezione oppure con la procedura descritta in *Aggiornamento di un client autonomo* a pagina 31.

Per eseguire l'installazione tipica di un client Java autonomo:

- 1 Chiudere tutte le applicazioni Windows, compreso ServiceCenter, prima di procedere all'installazione. Il client Java deve accedere ad alcune .dll condivise. La mancata chiusura di tutte le altre applicazioni Windows prima dell'installazione può causare errori in fase di connessione o all'apertura del client Java.

Per l'esecuzione, il client Java autonomo necessita di un JRE residente. Il CD-ROM di installazione di ServiceCenter contiene un JRE predefinito installabile oppure è possibile utilizzare un Java Runtime Environment (JRE) già installato, versione 1.2.2_008 o successiva. Per informazioni complete sui requisiti e la compatibilità della piattaforma corrente, vedere *Sito Web CenterPoint di Peregrine* a pagina 9. Una volta stabilita la versione supportata che si desidera utilizzare, è possibile installarla direttamente dal sito Web Sun. Per ulteriori informazioni, vedere *Installazione del Java Runtime Environment* a pagina 54.

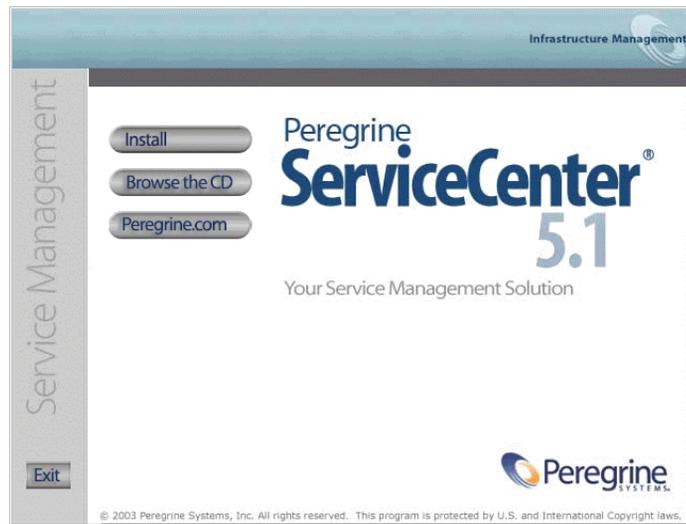
- 2 Per un client Java autonomo con un server di ServiceCenter su un'unica workstation, seguire la procedura relativa all'installazione tipica descritta nella *Guida all'installazione client/server per Windows*. Terminata l'installazione tipica, passare al capitolo successivo, *Configurazione del Client* a pagina 59.

Per eseguire l'installazione personalizzata di un client Java autonomo:

- 1 Introdurre il CD-ROM di installazione di ServiceCenter nell'unità corrispondente della propria workstation o server. Se l'installazione avviene in un sistema su cui è abilitata l'esecuzione automatica, il file setup.exe si avvia automaticamente. In alternativa, per avviare l'esecuzione è possibile scegliere uno dei seguenti metodi:
 - Passare alla directory del CD-ROM utilizzando Esplora risorse. Fare doppio clic su autorun.exe.
 - Avviare l'installazione di ServiceCenter dal prompt dei comandi di Windows. Digitare il seguente comando:

```
D:\>setup
```

dove D definisce l'unità CD-ROM.
- 2 Verrà visualizzata la schermata di avvio di Peregrine, come illustrato nella Figura 2-1. Fare clic su **Installa** per iniziare l'installazione.

**Figura 2-1: Schermata di avvio di ServiceCenter**

Viene visualizzato il messaggio illustrato nella Figura 2-2. Fare clic su **OK**. È possibile avere installazioni multiple del server di ServiceCenter sulla stessa workstation ed eseguirle contemporaneamente. Il programma di installazione prevede una nuova directory per ciascuna nuova istanza di ServiceCenter installata. Le versioni precedenti possono continuare a funzionare in directory separate, tuttavia ciascuna istanza deve avere un numero di porta diverso.



Figura 2-2: Avviso di installazioni multiple

- Viene avviata l'installazione guidata, come illustrato nella Figura 2-3. Fare clic su **Avanti**.

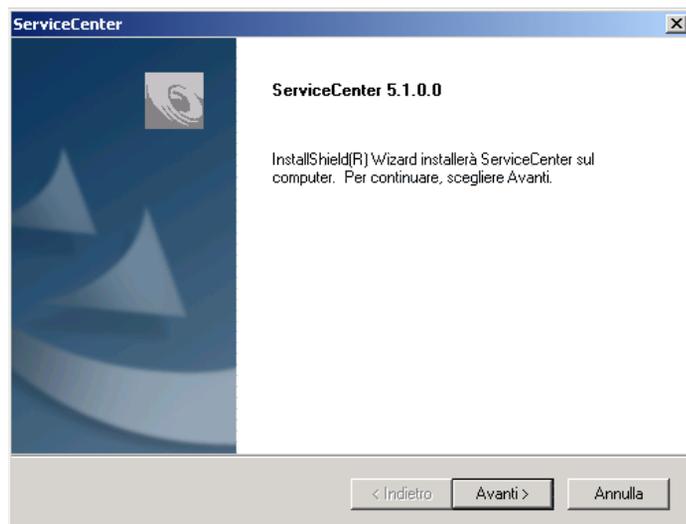


Figura 2-3: Installazione guidata

- 4 Quando viene visualizzata la finestra Tipo di installazione, illustrata nella Figura 2-4, selezionare **Personalizzata**. Fare clic su **Avanti**.

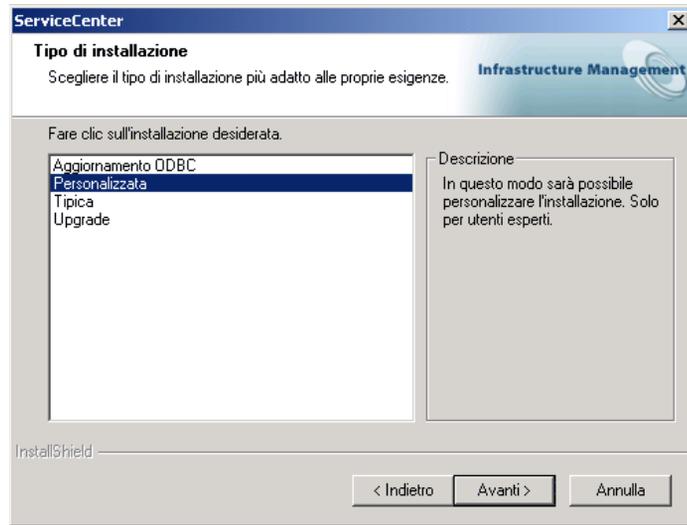


Figura 2-4: Finestra Tipo di installazione

Nota: Fare clic su **Indietro** per tornare a una finestra precedente e modificare i dati immessi. Fare clic su **Annulla** se si desidera interrompere del tutto l'installazione.

- 5 Verrà visualizzata la finestra Scegliere la posizione di destinazione. L'installazione tipica crea una cartella C:\Programmi\Peregrine\ServiceCenter. La Figura 2-5 mostra la cartella di destinazione predefinita. Fare clic su **Sfoglia** per specificare un percorso diverso. Fare clic su **Avanti**.

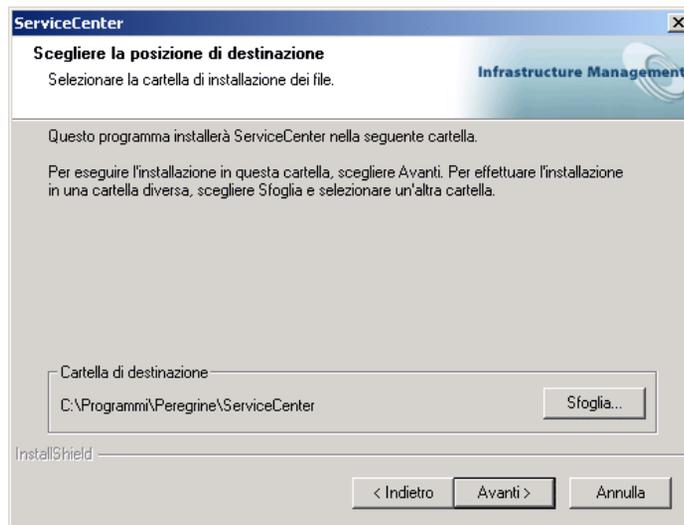


Figura 2-5: Finestra Scegliere la posizione di destinazione

Se si tratta di un'istanza multipla di un client Java, l'installazione assegna un nome di cartella univoco, ad esempio C:\Programmi\Peregrine\ServiceCenter2, oppure è possibile scegliere una cartella personalizzata.

Per installare un altro tipo di client Java in un'installazione preesistente, è necessario modificare tale installazione di ServiceCenter nel Pannello di controllo di Windows. Per ulteriori informazioni, vedere la *Guida all'installazione client/server per Windows*.

- 6 Viene visualizzata la finestra di Selezione componenti. Deselezionare tutte le caselle di controllo tranne **Client Java**, come illustrato nella Figura 2-6 a pagina 25. Fare clic su **Avanti**.

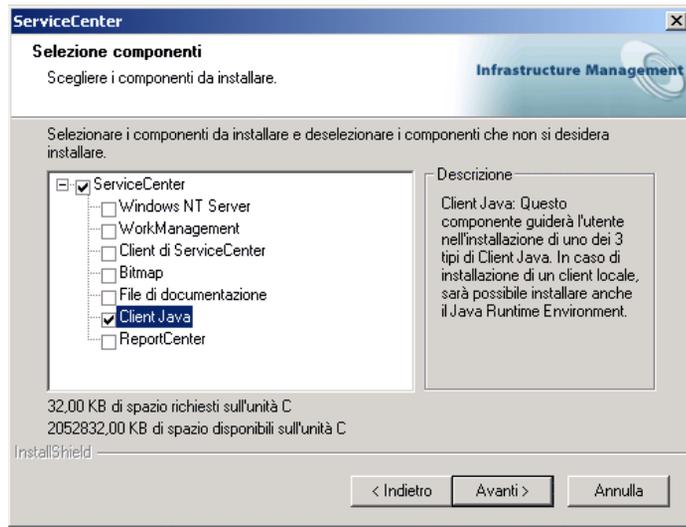


Figura 2-6: Finestra Selezione componenti per un client Java

- 7 La Figura 2-7 mostra la finestra Informazioni sulla licenza e Leggimi. Scorrendo queste informazioni si apprendono ulteriori dettagli sulle licenze, sull'accesso alle note sulla versione, sul sito Web CenterPoint di Peregrine e sull'assistenza clienti. Fare clic su **Avanti**.

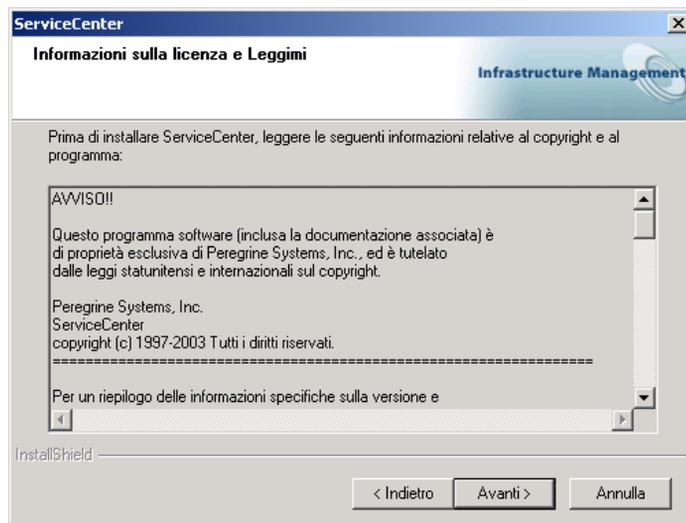


Figura 2-7: Finestra Informazioni sulla licenza e Leggimi

- 8 La Figura 2-8 mostra la finestra Informazioni sul server TCP/IP. L'installazione rileva automaticamente l'ID servizio assegnato (numero di porta) relativo alla workstation o al server. Se non vengono visualizzati numeri di porta, contattare l'amministratore del sistema per verificare la connessione TCP/IP. Fare clic su Avanti. L'installazione convalida il nome host.

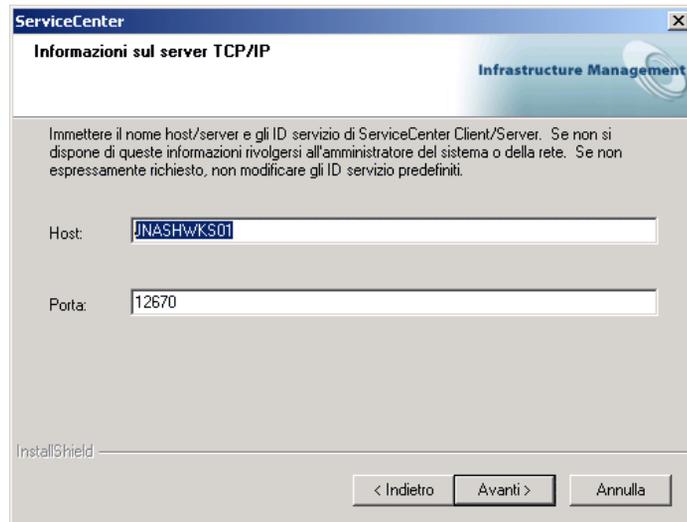


Figura 2-8: Informazioni sul server TCP/IP

Importante: Se il numero di porta predefinito è già assegnato a un altro server di ServiceCenter, è necessario scegliere un numero di porta diverso per evitare conflitti quando si eseguono istanze multiple di ServiceCenter.

- 9 Viene visualizzata la finestra Opzione client Java. La Figura 2-9 mostra l'installazione autonoma del client selezionata. Fare clic su **Avanti**.

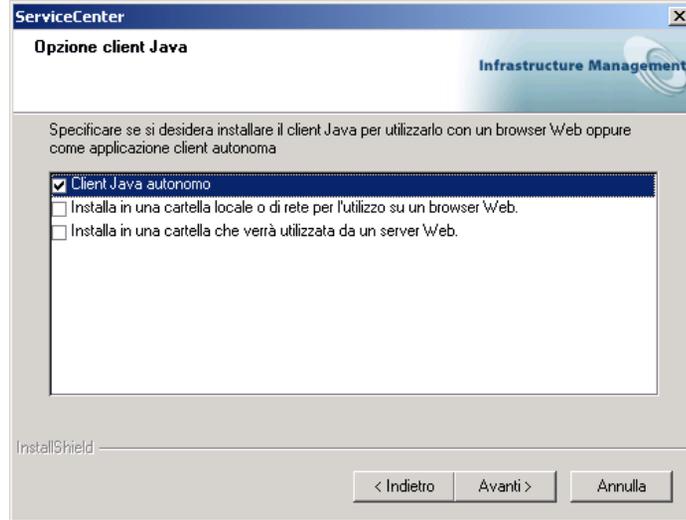


Figura 2-9: Opzione client Java

- 10 La Figura 2-10 mostra la finestra Java Runtime Environment.

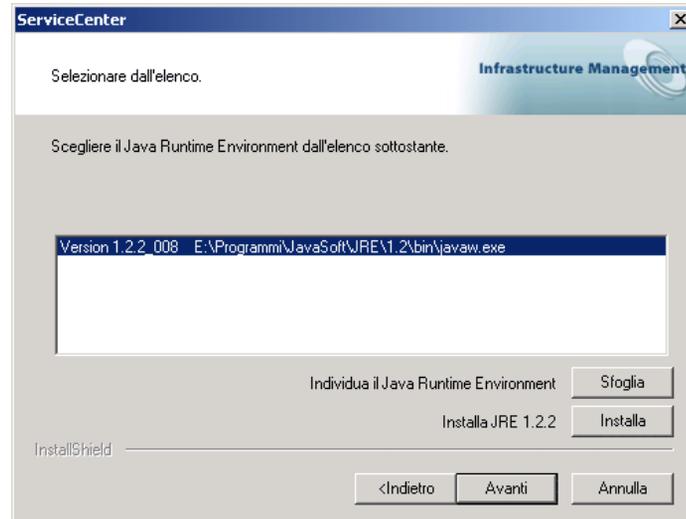


Figura 2-10: Finestra di selezione del JRE

Il client Java di ServiceCenter necessita di un JRE per l'esecuzione autonoma. Il JRE crea una Java Virtual Machine (JVM) utilizzata da questa applicazione autonoma. Se nel sistema sono installate più versioni JRE, selezionare dall'elenco la versione da utilizzare con ServiceCenter oppure, se non compare nell'elenco, individuarne il percorso.

Se nel sistema non è installato un JRE, fare clic su **Installa** per utilizzare la versione presente sul CD-ROM di installazione. Il JRE incluso in questa versione è Java 2 Runtime di Sun, versione 1.2.2_008. Se installata, questa versione diventa il JRE predefinito del sistema. Se la versione installata è diversa, prima di selezionarla verificare sul *Sito Web CenterPoint di Peregrine* che si tratti di una versione consigliata per ServiceCenter.

La Figura 2-11 mostra un JRE selezionato. Dopo aver selezionato un JRE esistente oppure aver installato la versione fornita con ServiceCenter, fare clic su **Avanti**.

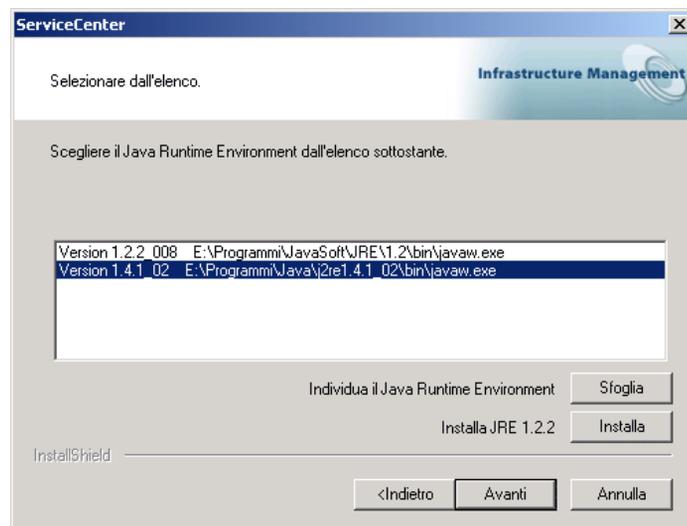


Figura 2-11: Finestra Java Runtime Environment

- 11 Viene visualizzata la finestra Cartella programmi, come illustrato nella Figura 2-12 a pagina 29. Il programma di installazione crea una nuova cartella di programmi ServiceCenter o consente di digitare un nome diverso per la cartella di programmi. Fare clic su **Avanti**.

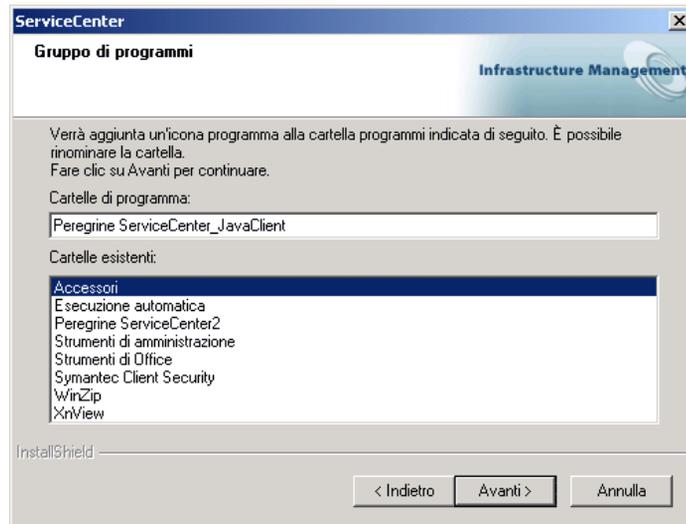


Figura 2-12: Cartella programmi

- 12 Il programma di installazione dispone di sufficienti informazioni per iniziare a copiare i file nella directory del programma specificata. La Figura 2-13 mostra un riepilogo delle impostazioni richieste durante il processo di installazione. Fare clic su **Avanti**.

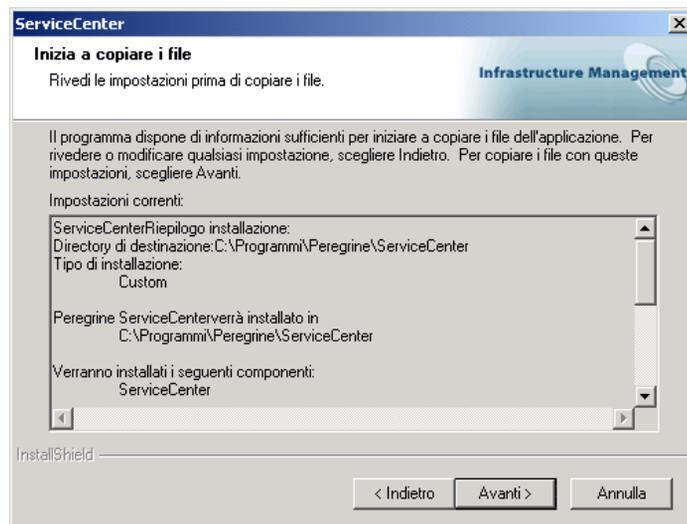


Figura 2-13: Finestra Copia dei file

- 13 L'installazione inizia a copiare i file selezionati, come illustrato nella Figura 2-14. È possibile interrompere l'installazione facendo clic su **Annulla**.

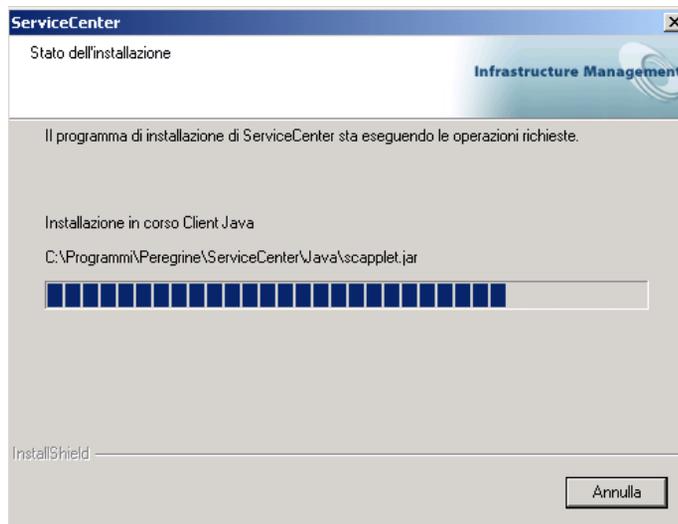


Figura 2-14: Finestra di stato dell'installazione

- 14 Al termine dell'installazione del client Java, viene visualizzata la finestra mostrata nella Figura 2-15.

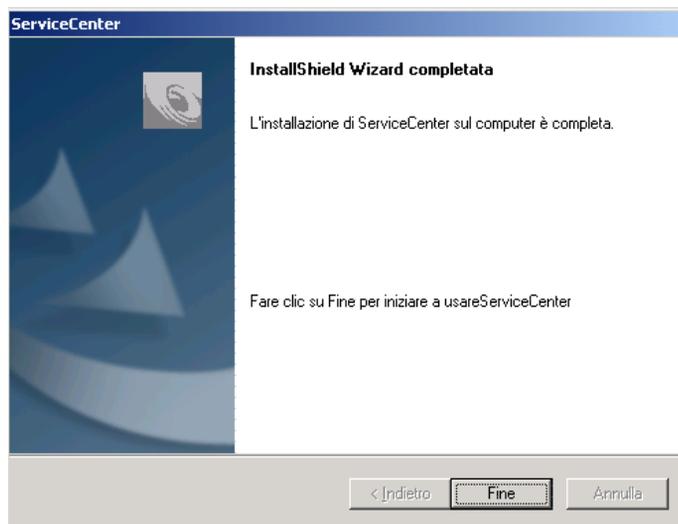


Figura 2-15: InstallShield Wizard completata

Esecuzione di un client autonomo

Per poter avviare il client Java, il server di ServiceCenter deve essere in esecuzione.

- 1 Per verificare lo stato del server, dal menu **Start** di Windows della workstation client selezionare **Programmi > Peregrine ServiceCenter > ServiceCenter Console**.
- 2 ServiceCenter Console indica con semaforo verde il server in esecuzione. Se il semaforo è rosso oppure giallo, fare clic su **Start** e attendere il semaforo verde.
- 3 Dal menu **Start** di Windows, selezionare **Programmi > Peregrine ServiceCenter > Client Java**. Verrà visualizzata la schermata di accesso a ServiceCenter.



Aggiornamento di un client autonomo

È possibile aggiornare o installare un client Java autonomo basato su browser anziché utilizzare il CD-ROM di installazione di ServiceCenter 5.1. Per procedere è necessario che ServiceCenter 5.1 sia installato sul server Web locale o remoto. Il client autonomo basato su browser comunica con l'installazione lato server per prelevare gli aggiornamenti.

Eeguire i passaggi indicati di seguito per l'aggiornamento o l'installazione:

- 1 Creare una copia del file `scjava.htm`.
- 2 Aprire la copia del file `..\java\scjava.htm` con un editor di testo e aggiungere i parametri seguenti
 - `<param name="InstallType" value="Standalone">`
 - `<param name="SCJ_Home" value=SERVICE_CENTER_JAVA_HOME>`
 - `<param name="JRE_Home" value=JAVA_HOME>`

dove `SERVICE_CENTER_JAVA_HOME` deve essere una directory sul sistema client. Ad esempio, `c:/scjavaclient` e `JAVA_HOME` sono i percorsi in cui è installato il JRE sul sistema client. Ad esempio:

`c:/programmi/javasoft/jdk1.3.1`

- 3 Modificare il valore del parametro ImagePath nel file .htm inserendo `SERVICE_CENTER_JAVA_HOME/bitmaps`.
- 4 Aprire il file modificato `scjava.htm` con un browser. All'apertura del file, si verificherà una delle seguenti situazioni:
 - Il client Java autonomo verrà installato nella directory specificata nel parametro `SCJ_Home`.
 - Se si tratta di una versione diversa, il client Java autonomo esistente installato nella directory specificata verrà aggiornato.
 - Se si tratta della stessa versione, il client Java autonomo esistente installato nella directory specificata verrà avviato.

Installazione di un client Java basato su browser

Quando il client Java viene eseguito come applicazione browser, il software del server di ServiceCenter può risiedere su un server locale oppure remoto. Il client può essere:

- Un client basato su browser locale rispetto al server Web su cui viene eseguito ServiceCenter
- Un client basato su browser remoto rispetto al server Web su cui viene eseguito ServiceCenter.

Per iniziare a utilizzare il client Java è necessario disporre dell'URL del server su cui si trova il file di avvio del client Java `scjava.launch.htm`.

Per verificare la compatibilità del browser, vedere *Sito Web CenterPoint di Peregrine* a pagina 9.

Client basato su browser locale

Questo client comunica solo con un server locale, pertanto non sono necessari URL.

Per installare un client Java basato su browser locale:

- 1 Per un client Java autonomo eseguire la procedura dal passaggio 1 al passaggio 8 iniziando da pagina 21.
- 2 Verrà visualizzata la finestra Opzione client Java. La Figura 2-16 a pagina 33 mostra il client browser Web selezionato. Fare clic su **Avanti**.

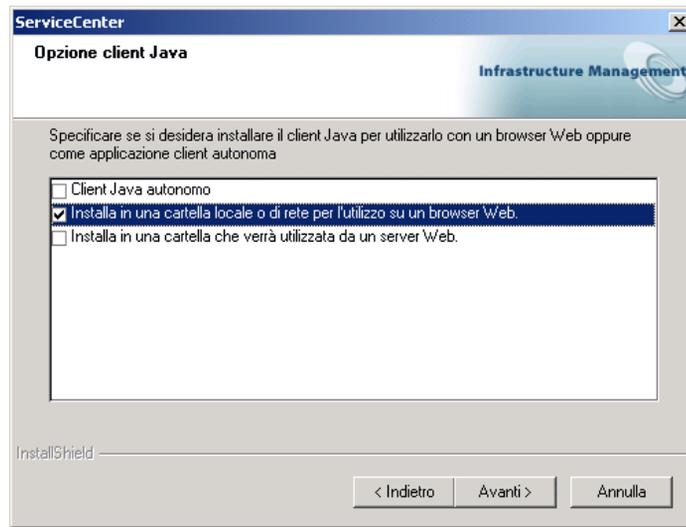


Figura 2-16: Opzione client browser

- 3 Tornare all'installazione del client Java iniziando dal passaggio 11 a pagina 28. Il passaggio 10 può essere ignorato. Il client Java basato su browser non richiede un JRE.

Per verificare la compatibilità del browser, vedere *Sito Web CenterPoint di Peregrine* a pagina 9.

Client basato su browser remoto

Questo client comunica solo con un server remoto. È necessario specificare un URL per effettuare la connessione remota.

Per installare un client Java basato su browser remoto:

- 1 Per un client Java autonomo eseguire la procedura dal passaggio 1 al passaggio 8 iniziando da pagina 21.
- 2 Verrà visualizzata la finestra Opzione client Java. La Figura 2-17 a pagina 34 mostra il client browser Web selezionato. Fare clic su **Avanti**.

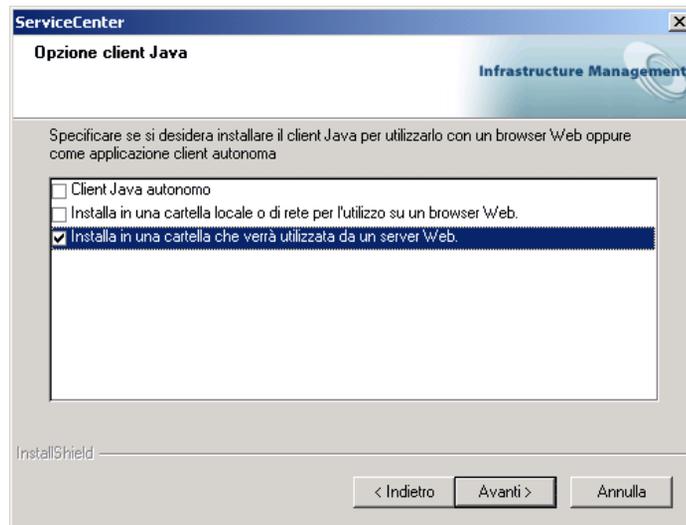


Figura 2-17: Opzione client Web

- 3 Verrà visualizzata la finestra Client Java. Digitare l'URL del server Web sul quale risiede il server di ServiceCenter. Specificando l'URL, non includere il prefisso `http://`. Seguire l'esempio mostrato nella Figura 2-18. Fare clic su **Avanti**.



Figura 2-18: URL del client Java

- 4 Tornare all'installazione del client Java iniziando dal passaggio 11 a pagina 28. Il passaggio 10 può essere ignorato. Il client Java basato su browser non richiede un JRE.

Esecuzione di un client Java basato su browser

Quando il client Java di ServiceCenter viene eseguito da un browser Web, il programma di installazione del client Java scarica una piccola applet che consente di eseguire il client, purché il browser indichi l'URL del server corretto.

Per eseguire un client Java basato su browser:

- 1 Dal menu **Start** di Windows, selezionare **Programmi, Peregrine ServiceCenter** e fare clic su **Client Java** con l'icona del browser. Il client Java di Windows utilizza l'icona di ServiceCenter. All'avvio, il client Java raccoglie le informazioni sull'ambiente per verificare che sia in esecuzione su una piattaforma e un browser supportati.

La Figura 2-19 mostra un avviso di protezione visualizzato. Fare clic su **Sì** per una singola sessione, oppure su **Sempre** per ignorare questo messaggio in futuro.

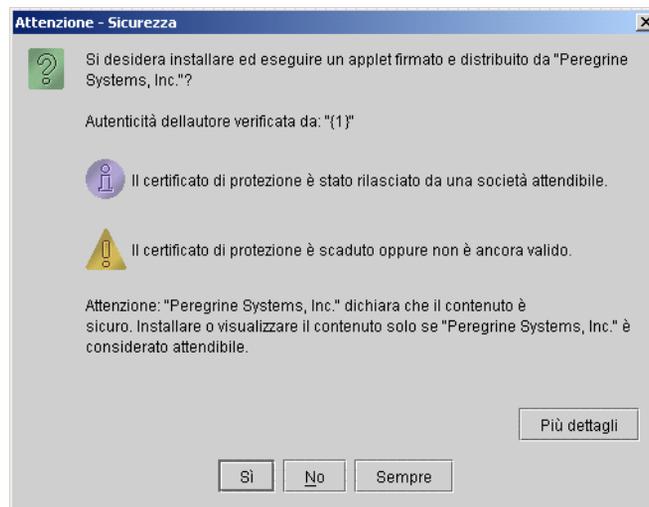


Figura 2-19: Avviso di protezione

Il client Java controlla se sul disco locale è presente una versione del client Java scaricato. Se è già presente un client Java aggiornato, il programma di installazione lo avvia e ha inizio una normale sessione di ServiceCenter.

- 2 Se non viene trovato alcun client Java, o se la versione del client è precedente a quella disponibile sul server, viene visualizzata una di due finestre. Dopo che l'utente ha scelto il tipo di download desiderato, il programma di installazione inizia a scaricare i file dell'applet del client. Al termine del download il client viene avviato. La Figura 2-20 mostra le opzioni disponibili. Fare clic su **Aggiorna** per scaricare i file dal server, oppure su **Continua** per utilizzare i file disponibili sulla workstation.



Figura 2-20: Opzioni di installazione

Nota: L'installazione del client Java viene avviata tramite la classe Java `com.peregrine.sc.installer.ClientInstaller`. L'applet del client Unix viene avviata tramite `com.peregrine.sc.client.ClientApplet`. Il client autonomo viene avviato tramite `com.peregrine.sc.client.ClientApplication`.

File scaricati

Il programma di installazione del Client Java esegue il download di alcuni file e li installa nel percorso della classe Java del browser. Gli utenti Netscape necessitano solo dei file immagine.

Nome file	Descrizione
Java client applet	Contiene il codice client di ServiceCenter (circa 1600 K).
Java client images	Contiene bitmap e altre immagini usate dal client (circa 600 K).

Nome file	Descrizione
Swing Library	Contiene la libreria Swing Java, vale a dire una serie di API GUI. In futuro alcuni browser potrebbero includere questa libreria, eliminando la necessità di scaricare il file (circa 2,4 MB).
scoicon.exe	Contiene immagini da visualizzare con allegati (circa 100 K).

File bitmap

Il programma di installazione del client Java di norma esegue il download dei file immagine e li memorizza nella cache. Vi è comunque la possibilità di specificare un parametro `imagepath` che indichi il percorso dei file non presenti nella directory cache. Utilizzare questo metodo per aggiungere file bitmap alle applicazioni di ServiceCenter e metterli a disposizione degli utenti del client Java.

Aggiungere i file bitmap con uno dei seguenti metodi:

- Aggiungere le immagini bitmap al file `bitmaps.zip` e aggiornare il file `scjversions.properties` nella directory del server Web.
- Inserire le immagini bitmap in una directory del server Web e includerne il percorso nel parametro `imagepath`.

File Htm

Per accedere al client Java tramite un browser, l'utente deve collegarsi a uno dei file della seguente tabella attraverso un URL e il percorso della directory del file. Ad esempio:

`http://vostroserver/java/scjavalaunch.htm`

dove *vostroserver* è il nome del server di ServiceCenter utilizzato.

Per ulteriori informazioni su questo tipo di configurazione, vedere *Client basati su browser* a pagina 15.

File Htm	Funzione
scapplet.htm	È utilizzato principalmente da alcuni sistemi UNIX che non sono compatibili con il meccanismo di firma usato dal client Java eseguito dal browser. Il file scapplet.htm utilizza un diverso meccanismo di firma. L'accesso a questo file è diretto, non tramite il file scjava13plugin.htm. Il file scapplet.htm utilizza automaticamente l'ultima versione del client individuata sul server. Il file funziona anche sui sistemi Windows.
scjava.htm	Utilizzare questo file su qualsiasi piattaforma (ad eccezione di Macintosh) compatibile con il meccanismo di firma standard del browser. Il file ha la funzione di scaricare automaticamente nuovi file client eventualmente individuati sul server. Accedendo a questo file direttamente (non tramite scjava13plugin.htm), il client verrà visualizzato in un browser, con i comandi del browser presenti. Per la connessione al server Web è necessario specificare l'URL del file scjava.htm.
scjava13plugin.htm	Questo file avvia ServiceCenter con JRE 1.3, se installato nel sistema. Se un utente si collega al file scjava13plugin.htm e il plug-in non è installato, il client Java richiederà di scaricare dal sito Web di Sun una versione supportata del JRE e di installarla.
scjava13plugin.htm	Utilizzare questo file per i sistemi Windows o Macintosh. Si tratta del file di connessione principale e contiene uno script Java che: <ul style="list-style-type: none"> • Determina se la piattaforma corrente è Windows o Macintosh • Apre in un browser il file .htm corretto nascondendo i comandi del browser.
scmac.htm	Utilizzare questo file per i sistemi Macintosh. Immettere nel browser il percorso del file scmac.htm residente su un server Web Windows o Unix accessibile. Le eventuali modifiche necessarie, ad esempio la configurazione del server hub, devono essere eseguite nel file scjavamac.htm. Solo per l'installazione locale: il sistema Macintosh supporta la versione autonoma del client Java. Accedere a questo file direttamente per avviare il client in un browser con i comandi del browser visualizzati.

Avvio di un client basato su browser

Per avviare il client Java è necessario che il server di ServiceCenter sia in esecuzione sul computer locale oppure sul server Web remoto.

Per eseguire il client Java in un browser

- ▶ Effettuare una delle seguenti operazioni:
 - Dal menu Start di Windows, selezionare **Programmi > Peregrine ServiceCenter > Java Client**.
 - Nel browser, digitare l'URL del file `scjava.launch.htm` o `scjava.htm` per la connessione al server Web. Ad esempio:
`http://vostroserver/java/scjava.launch.htm`
`http://vostroserver/java/scjava.htm`.

Il file `scjava.launch.htm` (o `scjava.htm`) apre il file `scjavamac.htm`, che viene visualizzato nel browser del client. Al momento della connessione, nel browser Web viene visualizzata la schermata di accesso di ServiceCenter.

Nota: È necessario che il server di ServiceCenter e il server Web al quale l'utente si collega siano entrambi avviati.

Modifica delle dimensioni heap del client Java

In un JRE Sun procedere nel modo seguente.

- 1 Dal menu Start di Windows, scegliere **Programmi > Peregrine ServiceCenter > Client Java** e fare clic con il pulsante destro del mouse.
- 2 Scegliere **Proprietà**.
- 3 Nel percorso di destinazione, anteporre `-Xms64 -Xmx64` al percorso della classe.

Sistemi operativi Unix

Il CD-ROM di installazione principale include l'installazione del client Java su Unix. Il client Java di ServiceCenter può essere installato e configurato come client autonomo oppure basato su browser. Prima di eseguire il programma di installazione, esaminare le opzioni offerte dai due tipi di installazione e scegliere quella che risponde meglio alle proprie esigenze. Per ulteriori informazioni, vedere *Considerazioni sull'installazione* a pagina 15.

Nota: Il client Java non può connettersi a un numero di porta maggiore di 65535.

Avviso: Non installare il client Java come root (superuser).

Prima dell'installazione:

- 1 Inserire il CD-ROM di ServiceCenter nella relativa unità.
- 2 Installare l'unità CD-ROM.
- 3 Creare una directory sotto la directory principale dei documenti del server Web (se si intende rendere il client Java disponibile come URL web), oppure creare una directory apposita in un altro percorso nel sistema. Se si crea questa directory come root, attribuire proprietà e autorizzazioni all'utente amministratore di ServiceCenter. È necessario disporre delle necessarie autorizzazioni per la creazione delle directory. Lo script di installazione tenterà di creare le directory specificate, se queste non esistono.
- 4 Verificare che la directory creata per l'installazione del client Java si trovi nel percorso del proprietario di ServiceCenter.
- 5 Cambiare la directory selezionando l'unità CD-ROM.

Per installare il client Java di ServiceCenter:

- 1 Passare alla directory Unix sul CD-ROM di installazione.
- 2 Avviare lo script di installazione eseguibile (install.sh).
- 3 Immettere la directory principale di installazione del client Java, vale a dire la directory in cui si desidera copiare i file del client Java. Ad esempio, digitando ServiceCenter i file del client Java risiederanno nella directory /ServiceCenter/java.

Nota: Il sistema esegue la convalida del nome di directory specificato. Se si indica una directory non valida, verrà generato un messaggio di errore. Se dopo tre tentativi l'installazione non è in grado di convalidare il nome di directory immesso, lo script di installazione verrà terminato e verrà generato un messaggio di errore.

- 4 Scegliere il prodotto da installare:
 - ServiceCenter (comprende il Client Java)
 - Client Java

Selezionare l'opzione 2, Client Java.

- 5 Seguire le istruzioni via via visualizzate, utilizzando le informazioni riportate di seguito per installare il client come applicazione autonoma oppure basata su browser.
 - **Installazione del client basato su browser**
Digitare `n` per installare il client Java come applicazione autonoma.
 - **Variabile servizio:** immettere il numero del servizio di ServiceCenter al quale si conetterà il client Java, ad esempio 12670. Immettere esclusivamente valori numerici. Se ServiceCenter è in esecuzione come servizio con nome (definito nel file `etc/services` del sistema) non specificare il nome del servizio. Digitare il numero di porta assegnato al servizio con nome.
 - **Installazione del client basato su browser**
Digitare `y` per installare il client Java come applicazione autonoma.
 - **Variabile codebase** - non utilizzata dall'installazione autonoma. Digitare l'URL della directory di installazione del client Java. Ad esempio, se si installa il client in una directory denominata `java` sotto la root dei documenti del server Web della propria società, la variabile codebase sarà `http://www.mycompany.com/java`. Non digitare **http://**.
 - **Variabile servizio** - immettere il numero del servizio di ServiceCenter al quale si conetterà il client Java, ad esempio 12670. Immettere esclusivamente valori numerici. Se ServiceCenter è in esecuzione come servizio con nome (definito nel file `etc/services` del sistema) non specificare il nome del servizio. Digitare il numero di porta assegnato al servizio con nome.

Esecuzione di un client autonomo

Una volta installato ServiceCenter su un sistema al quale il client Java è in grado di connettersi, sarà possibile eseguire il client Java come applicazione autonoma. Per poter eseguire il client Java come applicazione autonoma su una workstation Unix, il JRE deve essere installato nel percorso dell'utente.

Per eseguire il client Java di ServiceCenter:

- 1 Passare alla directory di installazione del client Java.
- 2 Passare alla directory /Run.
- 3 Digitare `scjava` nella riga di comando. Il client Java di ServiceCenter verrà avviato come applicazione.

Parametri della riga di comando

Nello script `scjava` è possibile specificare diversi parametri, tuttavia si raccomanda di farlo solo dopo aver acquisito una buona conoscenza dell'impostazione dei parametri. L'impostazione errata dei parametri comporterebbe errori di funzionamento del client Java.

Sintassi

Nella riga di comando digitare:

```
scjava -parametro argomento
```

Immettere il comando, il parametro e l'argomento per modificare l'impostazione predefinita.

Parametro	Descrizione ed esempi
-- <-AppArgs>	<p>Specifica altri argomenti non interpretati dallo script di avvio <code>scjava</code>. Lo script di avvio passa questi argomenti all'applicazione Java. Se si utilizza questo parametro, deve essere specificato per ultimo nella riga di comando. Utilizzare questo metodo per passare qualsiasi parametro valido del client Java. Ad esempio, per passare l'argomento di un'applicazione, digitare:</p> <pre>scjava -- -huburl http://webserver</pre> <p>Per passare l'argomento di un'ubicazione JRE standard seguito dall'argomento di un'applicazione:</p> <pre>scjava -java /bin/jre/ -- -huburl http://webserver</pre>

Parametro	Descrizione ed esempi
args	Specifica gli argomenti da passare alla JVM. Alcuni esempi comuni sono -mx e -ms per controllare l'allocazione dell'heap. Per verificare le opzioni supportate e la sintassi corretta, fare riferimento alla documentazione relativa alla JVM. Se si specificano vari argomenti JVM oppure un argomento JVM contenente spazi, indicarli tra virgolette.
java	Specifica il percorso della JVM.
host	Specifica il nome host della workstation o del server sul quale sono in esecuzione i servizi di ServiceCenter. Omettendo queste informazioni, lo script di avvio analizza il file HTML installato (scjava launch.htm) alla ricerca delle informazioni corrette.
-service	Specifica il numero del servizio (porta) usato dalla workstation o server host. In genere il numero del servizio è 12670. Omettendo queste informazioni, lo script di avvio analizza il file HTML installato (scjava launch.htm) alla ricerca delle informazioni corrette.
-images	Specifica l'URL delle immagini di ServiceCenter. Omettendo queste informazioni, il percorso predefinito assegnato alle immagini sarà lo stesso in cui si trova lo script di avvio.

Per visualizzare l'elenco dei parametri contenuti nello script, digitare quanto segue nella riga di comando:

```
scjava -help
```

Gli amministratori del sistema possono modificare lo script `scjava`. Leggere con attenzione i commenti presenti nel file prima di apportare eventuali modifiche. È buona norma eseguire una copia di backup del file prima di salvare le nuove modifiche.

Esecuzione di un client basato su browser

Per consentire agli utenti di eseguire il client Java dal browser su sistemi UNIX, gli amministratori del sistema devono effettuare una delle seguenti operazioni:

- Metodo 1: Eseguire l'avvio iniziale del client Java come utente root.
- Metodo 2: Modificare le autorizzazioni di accesso alla directory Netscape consentendo agli utenti pieno accesso.

Nota: Se gli utenti utilizzano sistemi operativi Windows o Macintosh per connettersi a un server Web Unix, questi passaggi possono essere ignorati.

Metodo 1:

- 1 Installare il client Java come utente normale.
- 2 Avviare il client Java come utente root in un browser utilizzando il file `scjava.launch.htm`. Il client verrà aggiornato automaticamente. In seguito gli utenti potranno avviare il client Java in un browser senza accedere come utente root. L'accesso come utente root sarà necessario unicamente per gli aggiornamenti al client.

Metodo 2:

- 1 Installare il client Java come utente normale.
- 2 Modificare ricorsivamente le autorizzazioni nella directory `java/classes` sotto la directory principale di Netscape per concedere l'accesso completo (lettura, scrittura, esecuzione) a un determinato utente o gruppo di utenti.

Avvio del Client

Specificare l'URL del file `scjava.launch.htm` o `scjava.htm` per la connessione al server Web. Ad esempio:

```
http://vostroserver/java/scjava.launch.htm.  
http://vostroserver/java/scjava.htm.
```

Per i sistemi Unix che non sono compatibili con il meccanismo di firma usato dal client Java eseguito dal browser, specificare l'URL del file `scapplet.htm` con cui connettersi al server Web. Ad esempio:

```
http://vostroserver/java/scapplet.htm.
```

Modifica delle dimensioni heap del client Java

Occorrerà incrementare le dimensioni minima e massima dell'heap Java se si ha la necessità di migliorare le prestazioni del client Java, oppure nel caso in cui nella console siano visualizzati messaggi di `OutOfMemoryException` dopo l'avvio del client.

Per modificare la dimensione heap Java:

- 1 Passare alla directory di installazione del client Java.
- 2 Passare alla directory `/RUN` del client Java.

Nota: La directory `RUN` del client Java è diversa dalla directory `RUN` di `ServiceCenter`. La directory `RUN` del client Java contiene un solo file, `scjava`.

- 3 Aprire il file `scjava` con un editor Unix, ad esempio `vi`.
- 4 Nella sezione `User Configurable Variables` del file `scjava` impostare le dimensioni minime (iniziali) dell'heap:

```
-ms[dimensioni][unità]
```

dove `[dimensioni]` è un numero intero e `[unità]` è **k** (kilobyte) o **m** (megabyte).

- 5 Specificare le dimensioni massime dell'heap nella stessa sezione:

```
-mx[dimensioni][unità]
```

Ad esempio, per impostare le dimensioni di heap iniziale e massima rispettivamente a 32 MB e 48 MB, digitare:

```
SCJ_JRE_ARGS="-ms32m -mx48m"
```

- 6 Salvare e chiudere il file `scjava`.

Sistemi operativi Macintosh

Il CD-ROM di installazione di ServiceCenter contiene file di installazione per entrambi i sistemi operativi Mac OS 9.x e Mac OS X.

Entrambe le opzioni sono disponibili dal menu di installazione iniziale.

L'installazione del client Java su un Macintosh prevede tre passaggi:

- 1 Installare un JRE Macintosh. Il sistema OSX ha già installato un JRE valido. Vedere *Installazione di MRJ per OS 9.x*.
- 2 Per installare il client Java, vedere *Installazione di un client Java per OS 9.x* a pagina 49 o *Installazione del client Java: OS X* a pagina 51.
- 3 Aggiornare il file `scj.ini` per aggiungere il nome dell'host e il numero di porta del server. Vedere *Modifica del file scj.ini* a pagina 50.

Installazione di MRJ per OS 9.x

Per ottenere prestazioni ottimali si consiglia di utilizzare MRJ versione 2.2.5 o successiva. È possibile migliorare le prestazioni anche aggiungendo altra memoria. Un MRJ valido può essere installato utilizzando il programma di installazione fornito nel CD-ROM del client Java di ServiceCenter per Macintosh. Il programma di installazione del client Java crea sul disco rigido una cartella per il client Java di ServiceCenter e una cartella per l'installazione di MRJ.

Se è già installata la versione corretta di MRJ, procedere con la sezione successiva, *Installazione di un client Java per OS 9.x* a pagina 49. Per ulteriori informazioni sui requisiti hardware della piattaforma, vedere *Sito Web CenterPoint di Peregrine* a pagina 9.

- 1 Inserire il CD-ROM di installazione del client Java nella relativa unità. La Figura 2-21 mostra la finestra di dialogo iniziale dell'installazione.

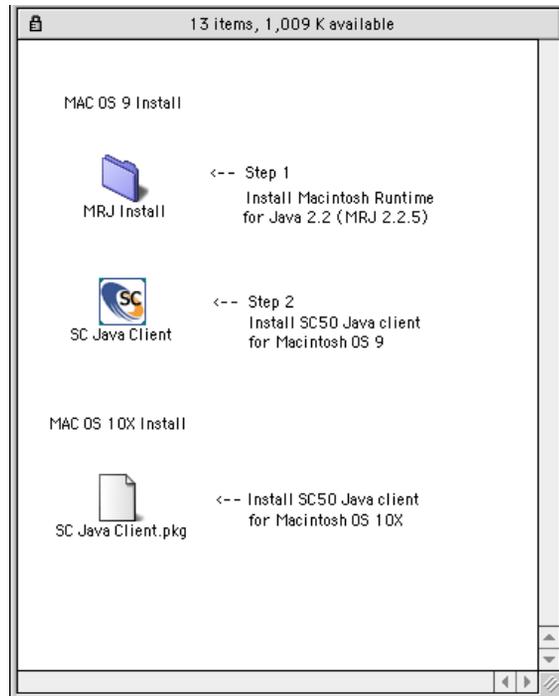


Figura 2-21: Finestra di installazione Macintosh

- 2 Fare doppio clic su **MRJ Install**. La Figura 2-22 mostra la finestra di installazione di MRJ.



Figura 2-22: Finestra di dialogo di l'installazione di MRJ

- 3 Fare doppio clic su **Installer**. Sarà visualizzato il contratto di licenza.

- 4 Per proseguire, fare clic su **Agree** (accetto) nella finestra di dialogo del contratto.

La Figura 2-23 mostra la finestra di dialogo in cui è possibile specificare il percorso di installazione. Fare clic su **Install** (installa) per confermare il percorso predefinito, oppure fare clic su **Switch Disk** (cambia disco) per selezionare un altro percorso, quindi fare clic su **Install** (installa). L'installazione verrà avviata.



Figura 2-23: Finestra di dialogo di installazione di MRJ

- 5 Quando viene visualizzato il messaggio che segnala il completamento dell'installazione di MRJ, fare clic su **Quit** (esci).

Verrà nuovamente visualizzata la finestra di dialogo iniziale dell'installazione, Figura 2-21 a pagina 47.

- 6 Proseguire con il passaggio 2 a pagina 49, *Installazione di un client Java per OS 9.x*. Non è necessario riavviare la workstation dopo l'installazione di MRJ.

Installazione di un client Java per OS 9.x

- 1 Inserire il CD-ROM di installazione del client Java nella relativa unità. La Figura 2-24 mostra la finestra di dialogo iniziale dell'installazione.

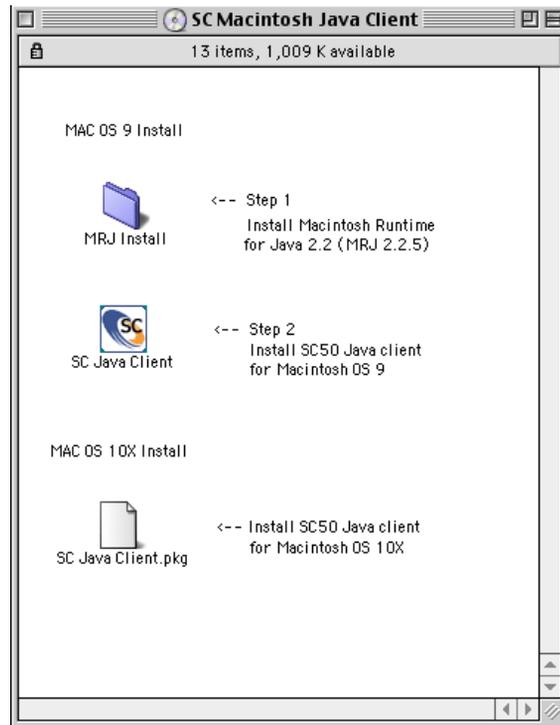


Figura 2-24: Finestra di installazione Macintosh

- 2 Fare doppio clic su **SC Java Client**. Verranno visualizzate le note sulla versione del client Java.
- 3 Dopo aver letto le note sulla versione fare clic su **Continue** (continua).

- 4 La Figura 2-25 mostra la finestra di dialogo Client Java di ServiceCenter in cui è possibile specificare il percorso di installazione. Per installare il client Java al livello root dell'unità di avvio fare clic su **Install** (installa). Per definire un percorso diverso, selezionare **Select Folder** (scegli cartella) dalla casella di riepilogo, quindi fare clic su **Install** (installa). L'installazione verrà avviata.



Figura 2-25: Finestra di dialogo Client Java di ServiceCenter

- 5 Quando viene visualizzato il messaggio che segnala il completamento dell'installazione del client Java, fare clic su **Quit** (esci).

Modifica del file scj.ini

- 1 Aprire la cartella del client Java di ServiceCenter mostrata nella Figura 2-26.

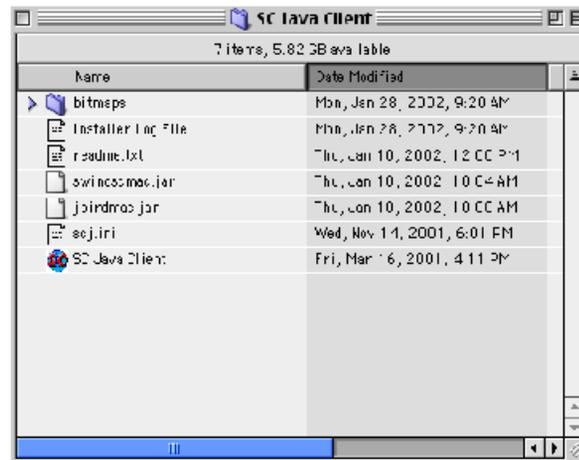


Figura 2-26: Cartella Client Java di ServiceCenter

- 2 Fare doppio clic sul file `scj.ini` per aprirlo con l'editor Apple SimpleText.
- 3 Modificare la riga 1. Eliminare **nome host** e sostituirlo con il nome host o l'indirizzo IP del server di destinazione di ServiceCenter.
- 4 Modificare la riga 2. Eliminare il numero di porta predefinito per la connessione al server, **service:12670**. Sostituirlo con il numero di porta corretto, ad esempio **service:12680**.
- 5 Dal menu File, scegliere **Save** (registra) per salvare le modifiche.
- 6 Dal menu File fare clic su **Quit** (esci) per uscire dall'editor SimpleText.
- 7 Copiare il file `scj.ini` nella cartella **Sistema > Preferenze**. Se si tralascia questo passaggio, il client Java non si avvierà in modalità autonoma.

Installazione del client Java: OS X

Seguire la procedura descritta in questa sezione per installare il client Java su una piattaforma con sistema operativo Macintosh OS X. Macintosh OS X dispone di un MRJ residente.

Per installare un client Java OS X:

- 1 Inserire il CD-ROM di installazione del client Java nella relativa unità. La Figura 2-27 mostra la schermata iniziale dell'installazione. Fare doppio clic su `SC Java client.pkg`.

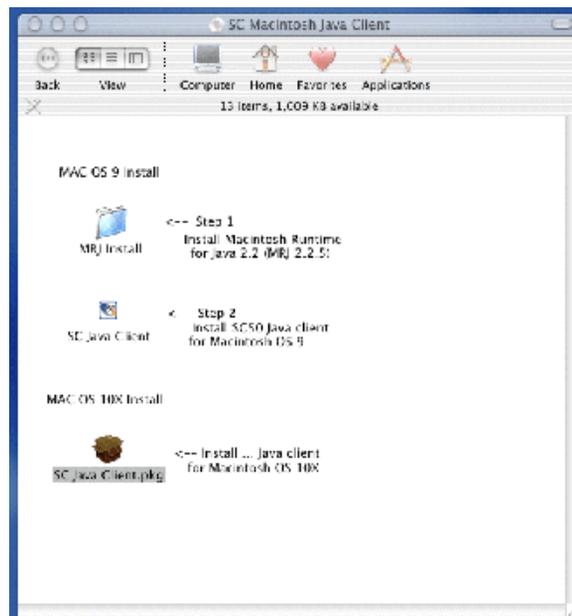


Figura 2-27: Menu di installazione Mac OS X

- 2 La Figura 2-28 mostra la finestra Select a Destination (scegli una destinazione). Selezionare il volume e la cartella in cui si desidera copiare il software. Se necessario, fare clic su **Choose...** (scegli...) per cambiare cartella. Fare clic su **Continue** (continue) per iniziare l'installazione.

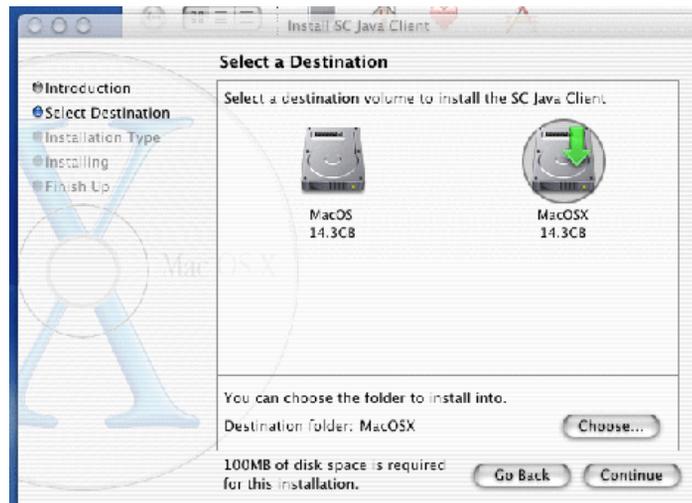


Figura 2-28: Scelta della destinazione

- 3 La Figura 2-29 mostra la finestra Easy Install (installazione standard). Fare clic su **Install** (installa).

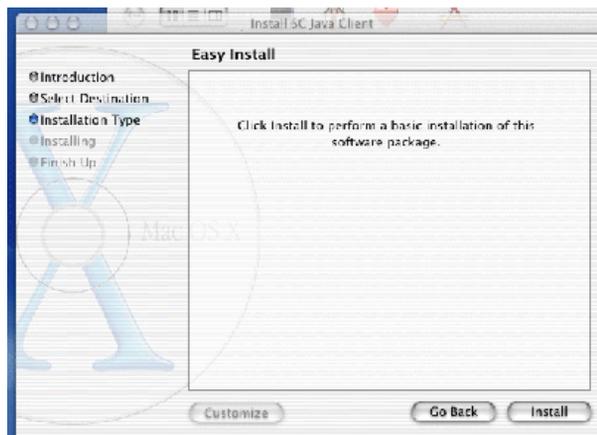


Figura 2-29: Finestra Easy Install (Installazione Standard)

- 4 La procedura guidata informerà l'utente del completamento dell'installazione. Fare clic su **Close** (chiudi).

Il client verrà installato in una directory chiamata **SC Java Client**. Aprire il file `sc.ini` da questa directory con un editor di testo. Seguire le procedure descritte in *Modifica del file `scj.ini`* a pagina 50.

Esecuzione di un client autonomo

Per avviare il client autonomo, fare doppio clic sull'icona SC Java client nella cartella SCJavaClient.

Esecuzione di un client basato su browser

Per eseguire il client Java in un browser su sistemi Macintosh è necessario connettersi a un server Web tramite un URL che indichi il server Web. Sul server Web è presente un file `scjavalaunch.htm` che apre il file `scjavamac.htm`. Nel browser del client viene visualizzato il file `scjavamac.htm`.

Ad esempio, è necessario avviare una sessione del browser su un sistema Macintosh indicante `http://vostroserver/percorso_di_scjavalaunch.htm` su un server remoto su cui è in esecuzione ServiceCenter. Il file `scjavalaunch.htm` a sua volta apre il file `scjavamac.htm`, che viene visualizzato nel browser del client.

Le eventuali modifiche necessarie, ad esempio la configurazione del server hub, devono essere eseguite nel file `scjavamac.htm`.

Sistemi operativi OS/2

Il sistema operativo OS/2 può essere utilizzato come host per un client Java di ServiceCenter, ma non per un server di ServiceCenter.

Per installare il client Java:

- 1 Inserire il CD-ROM in un'unità locale.
- 2 Sul CD-ROM individuare ed eseguire `../OS2/setup.exe`.
- 3 Durante il processo di installazione, specificare la directory in cui sarà installato il client Java.

Esecuzione di un client autonomo

- Aprire la console di OS/2 e digitare quanto segue su un'unica riga, sostituendo eventualmente la lettera associata all'unità.

Nota: È necessaria la versione 1.1.8 di JRE. Sostituire la directory di installazione con <jre directory>.

Il comando <spazio> indica dove immettere uno spazio. Non digitare <spazio> letteralmente.

```
C:\<jre directory>\bin\java<spazio>-classpath<spazio>
C:\<jre directory>\lib\classes.zip;C:\<javaclient
directory>\java\jbird.jar;C:\<javaclient directory>\java\swingsc.jar<space>
com.peregrine.sc.client.ClientApplication<spazio>
-host:<host><spazio>-service:<port><space>
-imagepath:C:\<javaclient directory>\java\bitmaps<invio>
```

Installazione del Java Runtime Environment

È possibile scaricare dal sito Web di Sun le versioni più recenti del JRE eseguendo il plug-in Get Java. Le prestazioni del JRE 1.3 sono superiori rispetto alle versioni precedenti. Quando si installa questa versione sul server impostandola come JVM predefinita, il client Java di ServiceCenter può connettersi al server di ServiceCenter utilizzando il file scjava.launch.htm standard.

Per installare il JRE:

- 1 Connettersi a <http://java.sun.com/getjava>.

- 2 Seguire le istruzioni per scaricare il plug-in Get Java. Vengono scaricati i file di configurazione e l'installazione si avvia automaticamente. Questo plug-in diventa il JRE predefinito del browser se si utilizza Internet Explorer. Anche Netscape versione 6 e successive utilizzano il plug-in Java come JRE predefinito. Ciò significa che i file HTML che avviano delle applet non richiedono modifiche per utilizzare direttamente il plug-in Java.



Se il client Java di ServiceCenter ha un JRE versione 1.3 con avvio tramite `scjava13plugin.htm`, il client Java utilizzerà il JRE locale versione 1.3. Se non si dispone del JRE locale 1.3, verrà visualizzata la richiesta di scaricare dal sito Web di Sun il JRE versione 1.3 e di installarlo.

Per utilizzare la versione JRE 1.3:

- 1 Installare la versione JRE 1.3 sul server.
- 2 Richiedere agli utenti la connessione a ServiceCenter tramite il file `scjava13plugin.htm`.

Se gli utenti non hanno il JRE versione 1.3 installato e si connettono al file standard `scjavalaunch.htm`, il client Java di ServiceCenter si avvierà utilizzando la JVM predefinita del browser.

Velocità di connessione

È possibile eseguire un test della connessione di rete tra il server di ServiceCenter e il client Java. Premere `Ctrl+Maiusc+S` per avviare un test che prevede l'invio di piccoli pacchetti di dati dal client al server. Il test ha una durata di circa 15 secondi. I risultati del test saranno visualizzati nella finestra Note attive, contenente il numero di trasmissioni server al secondo e la larghezza di banda approssimativa disponibile per il client Java. Tali informazioni sono disponibili anche nella console Java.

La Console Java

È possibile personalizzare le impostazioni di configurazione dal Pannello di controllo della workstation. È disponibile anche una finestra Console Java interattiva eseguibile dal browser.

Impostazioni della Console Java nel Pannello di controllo

Per visualizzare la Console Java:

- 1 Dal menu **Start**, fare clic su **Impostazioni** > **Pannello di controllo** > **Java Plug-in**.
- 2 Fare clic sulla scheda **Browser**. Il plug-in Java è il JRE predefinito per Microsoft Internet Explorer. Se si eseguono modifiche in questa scheda è necessario riavviare la workstation. La Figura 2-30 mostra la Console Java con il plug-in Java selezionato per Microsoft Internet Explorer.

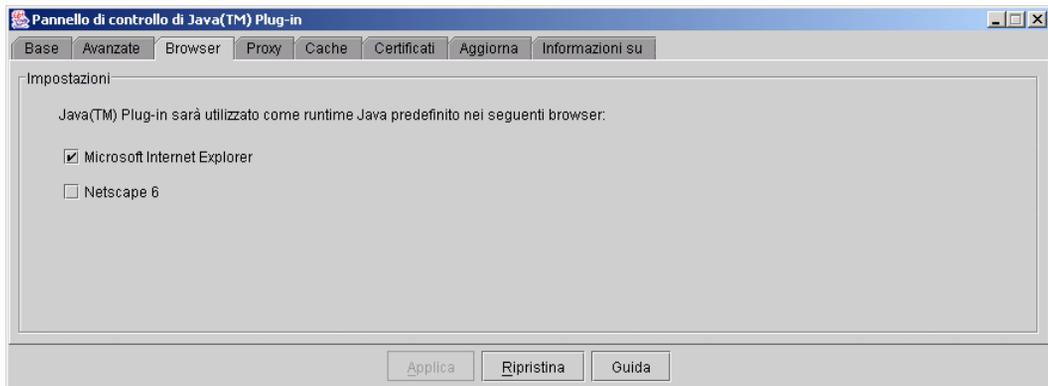
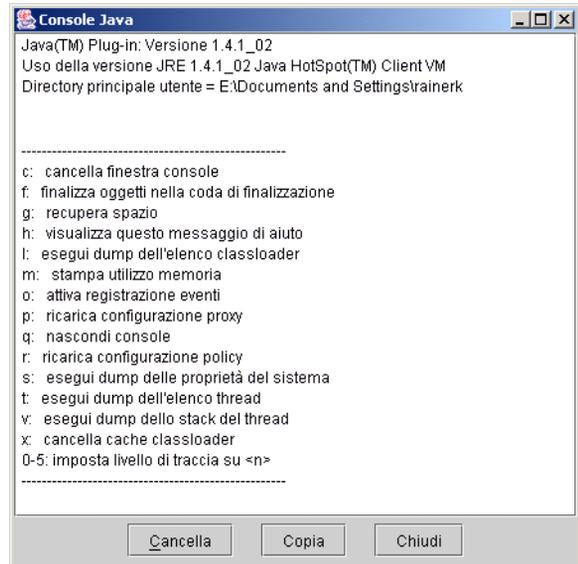


Figura 2-30: Console Java

Per visualizzare la finestra Console Java:

- 1 Avviare il browser.
- 2 Dal menu **Strumenti**, selezionare **Sun Java Console**. Verrà visualizzata l'icona Java (una tazzina di caffè) nella barra di sistema e la finestra Console Java con un elenco dei comandi utilizzabili durante la sessione del browser.



3 Configurazione del Client

CAPITOLO

Il corretto funzionamento del client Java nell'ambiente dell'utente si basa su determinati parametri e preferenze dell'utente. Tutti i nomi dei parametri sono impostati nel file `scjava.htm`, `scapplet.htm` o `scjavamac.htm`, a seconda del sistema in uso. È possibile passare i parametri dell'applicazione autonoma dalla riga di comando.

Leggere questo capitolo per ottenere informazioni su

- *File .htm del client Java* a pagina 60
- *Impostazione delle preferenze nei file .htm* a pagina 62
- *Impostazione delle preferenze in scj.ini* a pagina 63
- *Impostazione delle preferenze in scjpref.ini* a pagina 65
- *Impostazione delle preferenze in sc.ini* a pagina 66
- *Impostazione delle preferenze dalla riga di comando* a pagina 75
- *Codici delle lingue* a pagina 77

File .htm del client Java

Per eseguire il client Java da un browser, il browser carica un file HTML che richiama il codice della applet del client e fornisce parametri specifici necessari all'esecuzione, come quelli descritti nella sezione precedente. Per ulteriori informazioni, vedere *Esecuzione di un client Java basato su browser* a pagina 35.

L'esempio seguente illustra il codice della applet nel file scjava.htm.

```
<applet
codebase=http://www.company.com
code=com.peregrine.scinstaller.ClientInstaller
archive=sc.jar
width="100%"
height="100%"
vspace=0
hspace=0
align=middle>
<param name="cabbage" value="sc.cab">
<param name="Host" value="127.0.0.1">
<param name="Service" value="12670">
<param name="Language" value="en">
<param name="codeset" value="English">
<param name="Timeramount" value="15">
<param name="ImagePath" value="http://www.company.com/bitmaps/">
</applet>
```

Conoscendo i diversi parametri specificati in questo file è possibile apportare le modifiche necessarie. Le parti più significative di questo file sono descritte di seguito.

```
codebase=http://www.company.com
```

Specifica l'URL del percorso in cui risiedono i file del client Java. Nell'esempio è indicato il percorso di un server Web. Se si installano i file in un'unità locale, questo parametro deve contenere il percorso della directory. Ad esempio:

```
codebase="file:///C:\Programmi\...\Java"
<param name="Host" value="127.0.0.1">
```

Specifica l'indirizzo IP host del server di ServiceCenter. Questo indirizzo deve essere accessibile dai computer degli utenti. Se l'host fa parte di un firewall aziendale, solo gli utenti che rientrano in tale firewall saranno in grado di eseguire il client in modalità remota, a meno che non sia in esecuzione il componente Server Hub. Per ulteriori informazioni, vedere *Server Hub* a pagina 79. Digitare l'indirizzo IP del server a cui accedere. Il valore predefinito 127.0.0.1 indica l'host locale.

```
<param name="Service" value="12670">
```

La porta predefinita utilizzata dal server di ServiceCenter è 12670.

```
<param name="ImagePath" value="http://www.company.com/bitmaps/">
```

Specifica il percorso dei file immagine del client. In questo esempio è indicato il percorso di un server Web. Se si installano i file in un'unità locale, questo parametro deve contenere il percorso della directory. Ad esempio:

```
<param name="ImagePath" value="file:///C:\Programmi\...\bitmaps">
```

Durante l'installazione, le bitmap del client vengono scaricate automaticamente e installate sulla workstation remota del client. Questo parametro specifica una directory da utilizzare per le bitmap nel caso in cui non vengano individuati file bitmap sulla workstation del client Java.

Impostazione delle preferenze nei file .htm

È possibile specificare alcuni parametri opzionali nel file .htm utilizzato per avviare il client Java. Il formato di tutti i parametri è il seguente:

```
<param name="parametro" value="valore">
```

Nome parametro	Funzione
envdump	Genera un'immagine dell'ambiente del client Java a fini diagnostici. Quando si specifica questo parametro, il client crea l'immagine e termina immediatamente l'esecuzione.
codeset	Specifica la lingua da utilizzare. Specificare un identificatore ISO a due caratteri valido, ad esempio fr per francese. Per l'elenco completo, vedere <i>Codici delle lingue</i> a pagina 77.
language	Specifica la lingua, ad esempio inglese o giapponese, da utilizzare nella visualizzazione del client Java. È necessario specificare un identificatore ISO esteso, ad esempio French o English . Per l'elenco completo, vedere <i>Codici delle lingue</i> a pagina 77.
scjpath	Specifica il percorso del file scj.ini . Il file scj.ini è un file facoltativo che è possibile creare e salvare nella directory specificata da questo parametro. Per ulteriori informazioni, vedere <i>Impostazione delle preferenze in scj.ini</i> a pagina 63.
scjpref.ini	Quando è presente un file scjpath , viene generato un file scjpref.ini che memorizza le preferenze dell'utente.
DictionaryDir	Specifica la directory contenente i dizionari utilizzati dai correttori ortografici. Se non si desidera utilizzare i dizionari predefiniti forniti dal client, specificare nel parametro DictionaryDir il percorso della directory contenente il dizionario da utilizzare. Specificare un URL o un percorso assoluto.
DownloadDictionaryDir	Specifica una directory in cui scaricare i dizionari sulla propria workstation. Al termine del download del dizionario, il client Java scrive le informazioni relative alla directory nel file scj.ini . Specificare un URL o un percorso assoluto.

Parametri Dictionarydir e Downloaddictionarydir

Se sulla workstation in uso è presente un dizionario predefinito, il client Java lo utilizza per il controllo ortografico. Se il client non trova un dizionario locale, lo cerca con i seguenti metodi:

- Il client cerca nel percorso specificato dal parametro `dictionarydir`. Se il client trova un dizionario nel percorso specificato, lo utilizza per il controllo ortografico.
- Se non trova un dizionario nel percorso specificato dal parametro `dictionarydir`, il client cerca nel parametro `downloaddictionarydir` il percorso per scaricare i file di dizionario. Al termine del download, il sistema scrive il percorso del dizionario locale nel file `scj.ini` per evitare che il client scarichi nuovamente i file.

Impostazione delle preferenze in `scj.ini`

È possibile specificare i parametri nella riga di comando, nel file `.htm` o nel file `scj.ini`. Il file `scj.ini` è un file facoltativo che è possibile creare e salvare nella directory specificata dal parametro `scjpath`. Se individua il file `scj.ini`, per avviare il client l'applet utilizza i parametri in esso contenuti anziché quelli presenti nel file `.htm`. Una descrizione del parametro `scjpath` è fornita in *Impostazione delle preferenze nei file .htm* a pagina 62.

Se l'utente crea un file `scj.ini`, questo deve essere salvato nella directory specificata nelle preferenze utente. La directory di installazione predefinita è `C:\Programmi\Peregrine\ServiceCenter\Java`. La directory delle preferenze utente per il sistema in uso può essere individuata facendo clic su **Guida > Informazioni su ServiceCenter**. La Figura 3-1 mostra il percorso del file `scj.ini`.

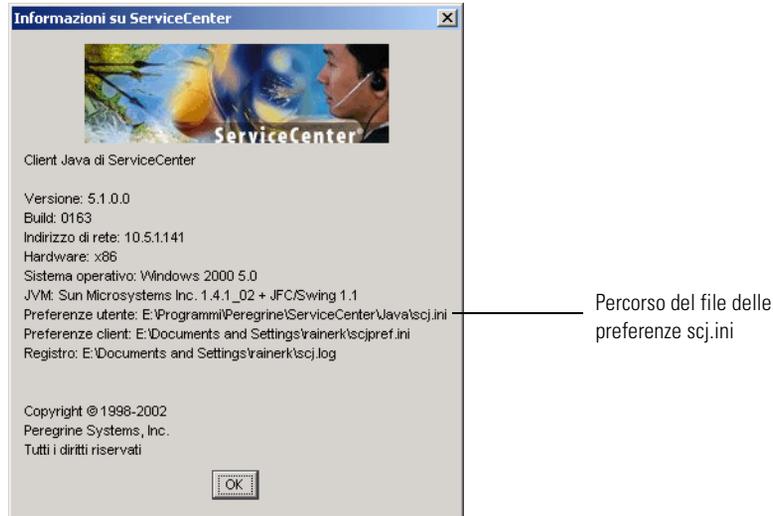


Figura 3-1: Directory delle preferenze utente

Il client Java utilizza i parametri contenuti in questo file solo se non sono stati specificati parametri nella riga di comando.

Formato del file `scj.ini`

Il formato di tutti i parametri nel file `scj.ini` è il seguente:

`nome:valore`

Le righe che iniziano con un carattere cancelletto (`#`) sono commenti.

Ad esempio:

```
# Questo è un commento
host:www.mycompany.com
service:12345
imagepath:http://www.mycompany.com/images
```

Altri parametri scj.ini

È possibile reindirizzare il client Java aggiungendo le variabili `host` e `service` nel file `scj.ini`. Il file `scj.ini` ha priorità rispetto alle impostazioni contenute nel file di avvio `.htm`. Se il file di avvio `.htm` contiene il parametro `scjpath`, per la connessione il client Java utilizza le variabili `host` e `service` specificate nel file `scj.ini`.

Parametro	Definizione
<code>host</code>	Specifica il nome host o l'indirizzo IP dell'host.
<code>service</code>	Specifica il nome o il numero del servizio di ServiceCenter per la connessione del client Java. Ad esempio, l'ID servizio predefinito è 12670.

Impostazione delle preferenze in scjpref.ini

Questo file contiene impostazioni di preferenze interne usate dal client Java. Il client Java legge ed eventualmente aggiorna questo file in fase di esecuzione. Il client Java utilizza questo file per memorizzare informazioni di esecuzione quali la posizione delle finestre, le preferenze relative ai tipi di carattere e ai colori, le dimensioni delle colonne delle tabelle e così via. La directory in cui è memorizzato questo file dipende dalla piattaforma in uso e dall'installazione. La directory delle preferenze utente per il sistema in uso può essere individuata facendo clic su **Guida > Informazioni su ServiceCenter**.

La Figura 3-2 mostra il percorso del file scjpref.ini.

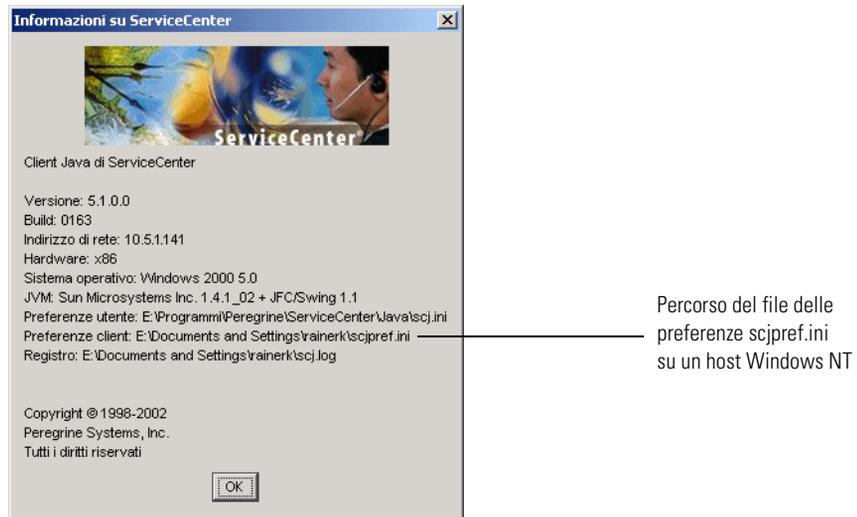


Figura 3-2: Directory delle preferenze client

Impostazione delle preferenze in sc.ini

Il server di ServiceCenter può gestire alcune preferenze del client Java con valori contenuti nel file `sc.ini` del server stesso. Se si specifica un valore non riconoscibile dal server, per impostazione predefinita viene utilizzato il valore 1. Se un parametro del file `sc.ini` è senza valore, il server utilizza il valore corrispondente eventualmente specificato nel file `scjpref.ini`.

Se si apporta una modifica a un file `.ini` di ServiceCenter, è necessario riavviare il server per attivarla.

La sintassi per questi parametri è la seguente: `parametro:variabile`

Parametro	Variabile
clientprinting	Specificare 0 oppure <i>n</i> , dove <i>n</i> è un numero positivo maggiore di zero.
explorer	Specificare 0 per disattivare ServiceCenter Explorer, 1 per attivarlo. Per ulteriori informazioni, vedere <i>Parametro explorer</i> a pagina 68.

Parametro	Variabile
<code>explorerdefault</code>	Specificare 0 perché all'avvio venga visualizzata la modalità Menu. Specificare 1 per la modalità SC Explorer. Nella modalità menu, è possibile selezionare SC Explorer dal menu Visualizza. All'avvio di una nuova installazione, la vista predefinita è la modalità SC Explorer. Per ulteriori informazioni, vedere <i>Parametro explorerdefault</i> a pagina 69.
<code>explorerhome</code>	Specificare il nome del menu iniziale se si crea un menu personalizzato. Il valore predefinito è <i>string1</i> . Se si specifica un valore non valido, ServiceCenter lo ignora e utilizza il valore predefinito. Per ulteriori informazioni, vedere <i>Esempio: explorerhome</i> a pagina 70. Ad esempio, se si crea un nuovo menu denominato java, specificare: <code>explorerhome:java</code>
<code>menuforms</code>	Specificare 0 per disattivare i moduli di menu, 1 per attivarli. Se i valori <code>explorer</code> e <code>menuforms</code> sono gli stessi, ad esempio 0 e 0 oppure 1 e 1, ServiceCenter ignora i due valori e utilizza l'ultima impostazione di visualizzazione. Peregrine Systems consiglia di specificare 1 per il parametro che si desidera utilizzare e 0 per l'altro parametro.
<code>nohelpnfield</code>	Specificare 0 per disattivare la funzione di guida, 1 per attivarla.
<code>sctimeramount</code>	Specificare <i>n</i> , dove <i>n</i> è un numero positivo maggiore di zero. Ad esempio, se si desidera che il server verifichi ciascun client ogni 20 secondi, impostare il parametro su 20 nel file <code>sc.ini</code> : <code>sctimeramount:20</code>
<code>viewactivenotes</code>	Specificare 0 per disattivare la funzione Note attive, 1 per attivarla.
<code>viewattachments</code>	Specificare 0 per disattivare gli allegati, 1 per visualizzare gli allegati. Gli allegati restano con il record, ma gli utenti non possono visualizzare o allegare file se questo parametro è disattivato.
<code>viewpromptforsave</code>	Specificare 0 per disattivare la richiesta di salvataggio del record quando si effettuano modifiche, oppure 1 per attivarla.
<code>viewrecordlist</code>	Specificare 0 per disattivare gli elenchi di record, 1 per visualizzare tali elenchi.

Parametro	Variabile
viewtoolbar	Specificare 0 per nascondere la barra degli strumenti comune, 1 per visualizzarla.
viewtraycaptions	Specificare 0 per disattivare le etichette dei pulsanti della barra di sistema, 1 per visualizzarle.

Parametro explorer

Le impostazioni nei file `sc.ini` e `scjpref.ini` possono influenzare la modalità di visualizzazione del client Java all'avvio.

Situazione ipotetica 1

Nel file `sc.ini` non è specificato alcun parametro `explorer`.

Se il file `scjpref.ini` esiste, all'avvio verranno visualizzate le ultime viste selezionate in quanto il client Java salva in questo file l'ultima impostazione selezionata. È possibile modificare manualmente la modalità di visualizzazione selezionando o deselezionando `SC Explorer` dal menu `Visualizza`.

Se il file `scjpref.ini` non esiste, per impostazione predefinita viene visualizzata la vista `SC Explorer`. È possibile modificare manualmente la modalità di visualizzazione selezionando o deselezionando `SC Explorer` dal menu `Visualizza`.

Situazione ipotetica 2

Il file `sc.ini` contiene questo parametro: `explorer:0`.

A prescindere dall'esistenza del file `scjpref.ini`, verrà visualizzata solo la vista `Moduli di menu`. Il menu `Visualizza` non conterrà l'opzione `SC Explorer`.

Situazione ipotetica 3

Il file `sc.ini` contiene questo parametro: `explorer:1`.

A prescindere dall'esistenza del file `scjpref.ini`, verrà visualizzata solo la vista `Explorer`. Il menu `Visualizza` non conterrà l'opzione `SC Explorer`.

Parametro explorerdefault

Le impostazioni nei file `sc.ini` e `scjpref.ini` possono influenzare la modalità di visualizzazione del client Java all'avvio.

Situazione ipotetica 1

Nel file `sc.ini` non è specificato alcun parametro `explorerdefault`.

Se il file `scjpref.ini` esiste, all'avvio verranno visualizzate le ultime viste selezionate in quanto il client Java salva in questo file l'ultima impostazione selezionata. È possibile modificare manualmente la modalità di visualizzazione selezionando o deselezionando **SC Explorer** dal menu **Visualizza**.

Se il file `scjpref.ini` non esiste, per impostazione predefinita all'avvio viene visualizzata la vista **SC Explorer**. È possibile modificare manualmente la modalità di visualizzazione selezionando o deselezionando **SC Explorer** dal menu **Visualizza**.

Situazione ipotetica 2

Il file `sc.ini` contiene questo parametro: `explorerdefault:0`.

A prescindere dall'esistenza del file `scjpref.ini`, all'avvio verrà visualizzata la vista **Moduli di menu** in quanto le impostazioni del file `sc.ini` hanno la priorità su qualsiasi altra impostazione. È possibile modificare manualmente la modalità di visualizzazione selezionando o deselezionando **SC Explorer** dal menu **Visualizza**.

Situazione ipotetica 3

Il file `sc.ini` contiene questo parametro: `explorerdefault:1`.

A prescindere dall'esistenza del file `scjpref.ini`, all'avvio verrà visualizzata la vista **SC Explorer** in quanto le impostazioni del file `sc.ini` hanno la priorità su qualsiasi altra impostazione. È possibile modificare manualmente la modalità di visualizzazione selezionando o deselezionando **SC Explorer** dal menu **Visualizza**.

Esempio: clientprinting

Se si specifica `clientprinting:0`, tutti i client connessi al server utilizzeranno la stampa da server. L'opzione Stampa sul client non sarà disponibile nella barra degli strumenti del client Java. Sarà selezionata l'opzione Stampa sul server.

Se si specifica `clientprinting:10`, tutti i client connessi al server utilizzeranno la stampante client e il limite di pagine del report client sarà di 10 pagine. Nella barra degli strumenti del client Java sarà selezionata l'opzione Stampa sul client. L'opzione Stampa sul server non sarà disponibile.

Se si impostano questi parametri tramite il server, le corrispondenti opzioni di menu saranno disattivate su tutti i client, non solo sul client Java. I client non possono ignorare le impostazioni del server.

Nota: Il client Java non supporta la limitazione dei report del client a n pagine per una stampa in corso.

Esempio: explorerhome

Gli amministratori del sistema possono personalizzare il menu di ServiceCenter Explorer creando un nuovo menu di avvio, modificando il record dell'operatore affinché rimandi al nuovo menu di avvio e aggiungendo il parametro `explorerhome` al file `sc.ini`. Questi passaggi possono essere eseguiti tramite lo stesso client Java, riavviandolo per visualizzare i risultati. I passaggi di base sono i seguenti:

- Creare un nuovo menu di livello root.
- Creare un record di menu.
- Aggiungere il parametro `explorerhome` nel file `sc.ini`.

Per creare un nuovo menu di livello root:

- 1 Dal menu principale di ServiceCenter, fare clic sulla scheda **Utilità**.
- 2 Fare clic sul pulsante **Amministrazione**. Verrà visualizzata la finestra di amministrazione.
- 3 Fare clic su **Operatori** nella casella di gruppo **Protezione**.

- 4 Immettere il nome di accesso (ad esempio falcon) del record operatore che si desidera modificare. Premere **Invio**. La Figura 3-3 mostra il record operatore.

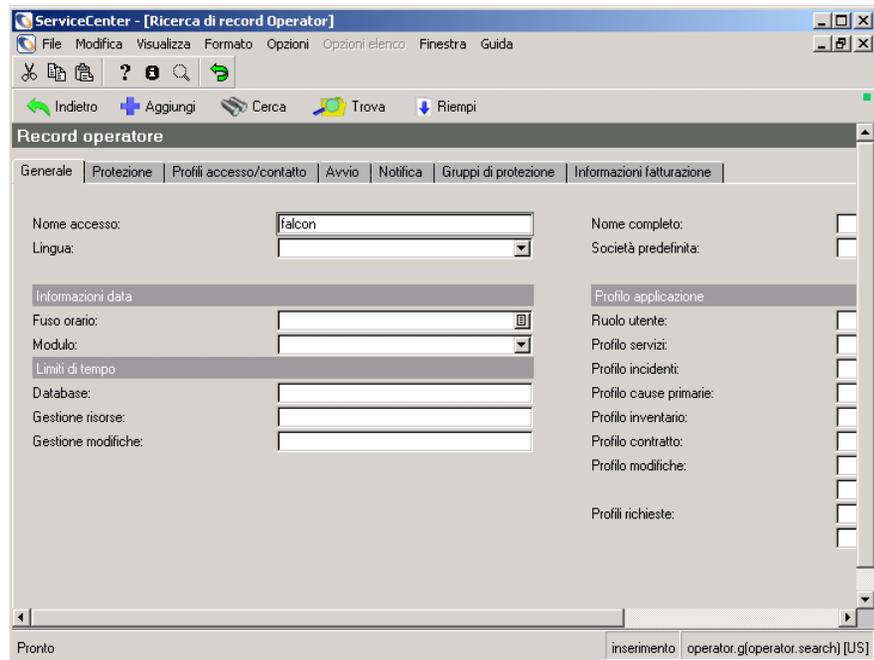


Figura 3-3: Ricerca di record operatore

- 5 Fare clic sulla scheda **Avvio**. Specificare un menu di avvio nella casella di gruppo **Applicazione iniziale**.
- 6 Nel primo campo disponibile immettere un nuovo **nome parametro**, ad esempio **java**.
- 7 Immettere un valore nel campo **Valori parametri**, ad esempio **JAVA HOME**. La Figura 3-4 a pagina 72 mostra l'area **Applicazione iniziale** con un nuovo nome parametro e valore parametro.

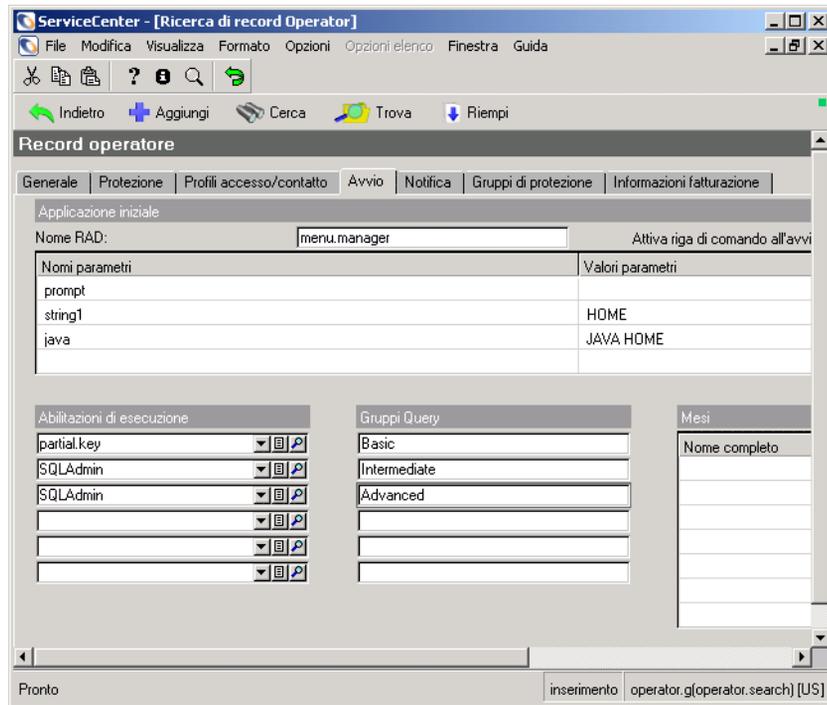


Figura 3-4: Informazioni in Applicazione iniziale

Avviso: I nomi di parametro name e string1 sono disponibili nei sistemi preconfigurati e non devono essere modificati. La modifica dei valori impostati nei parametri name o string1 può causare danni al sistema.

- 8 Fare clic su Salva.
 - Per creare un record di menu:**
 - 1 Dal menu principale di ServiceCenter, fare clic sulla scheda **Utilità**.
 - 2 Fare clic sul pulsante **Strumenti**.
 - 3 Fare clic sul pulsante **Menu**.

N. i	Descrizione	Comando	Applicazione	Nome param.	Valore parametro	Thread	Condizione
2	Services::Inventory Manag		menu.manage	name	ICM MAIN	true	lioption("Inventory Management")
3	Logout	Logout	menu.manage	name	LOGOUT		true
5	Services::Change Manage		menu.manage	name	CM3	true	lioption("Change Management") and not
6	Services::Request Manage		menu.manage	name	RM	true	lioption("Request Management")
7	Toolkit::Database Manage		database			true	true
8	Toolkit::Forms Designer		forms.designer			true	true
9	Toolkit::Report Writer		report.writer			true	true
10	Utilities::SQL Utilities		menu.manage	name	SQL GUI MENU	true	index("SysAdmin", \$lo.ucapex)>0 and inc
11	Utilities::Administration		menu.manage	name	ADMIN	true	index("SysAdmin", \$lo.ucapex)>0
12	Utilities::Event Services		menu.manage	name	ES	true	true
13	Toolkit::RAD Editor		encl.appl			true	index("SysAdmin", \$lo.ucapex)>0 or inde
14	Utilities::Maintenance		menu.manage	name	MAINT	true	index("SysAdmin", \$lo.ucapex)>0

Riga selezionata: 1 di 1 record

inserimento menu.g\db.view [US]

Figura 3-6: Informazioni in Applicazione iniziale

- 6 È possibile modificare qualsiasi campo nella colonna **Descrizione** per cambiare il testo visualizzato nel menu di SC Explorer.
- 7 È possibile modificare qualsiasi campo nella colonna **Comando** per cambiare il funzionamento del comando associato.
- 8 Fare clic su **Salva**.

Per aggiungere il parametro `explorerhome` nel file `sc.ini`:

- 1 Dal menu **Start** di Windows, scegliere **Programmi > Peregrine ServiceCenter > File di inizializzazione**. L'editor di testo predefinito aprirà `\\Peregrine\ServiceCenter\RUN\sc.ini`.
- 2 Aggiungere o modificare il parametro `explorerhome` :

```
explorerhome:java
```

Il valore del parametro `explorerhome` è quello specificato al passaggio 7 a pagina 71.

- 3 Salvare le modifiche e chiudere il file.

Note

Per modificare un ramo del menu Explorer, ripetere questa procedura per il menu del ramo.

- Per creare un nuovo menu di livello root vedere *pagina 70*.
- Per creare un record di menu vedere *pagina 72*.
- Per aggiungere il parametro `explorerhome` nel file `sc.ini` vedere *pagina 74*.

Ad esempio, modificare il menu SLA immettendo SLA nel campo di ricerca dei menu, quindi cambiando i valori del menu. Peregrine Systems consiglia di rinominare e aggiungere i nuovi menu prima di modificarli.

Impostazione delle preferenze dalla riga di comando

Quando si esegue il client Java come applicazione autonoma, viene richiamato il JRE installato nel sistema per l'esecuzione delle classi del client di ServiceCenter. È possibile passare all'applicazione autonoma la maggior parte dei parametri della applet tramite la riga di comando.

È possibile avviare il client Java autonomo con questo comando:

```
c:\path_to_jre -classpath c:\path_to_library\classes.zip;bird.jar;swingall.jar
com.peregrine.sc.client.Client Application -host:127.0.0.1 -service:12670
-imagePath:c:\path_to\BITMAPS
```

dove `classpath` indica i file della libreria delle classi per la libreria di runtime del JRE.

Questo comando avvia il JRE e carica la classe principale di ServiceCenter (`com.peregrine.sc.client.ClientApplication`) per l'esecuzione. Gli ultimi tre parametri sono obbligatori

Parametro	Definizione
<code>-host:host_ip</code>	Specifica l'indirizzo IP host del server di ServiceCenter. Questo indirizzo deve essere accessibile dalla workstation. Se l'host si trova all'interno di un firewall aziendale, solo gli utenti che rientrano nel firewall saranno in grado di eseguire il client.
<code>-service:service</code>	Specifica il servizio usato dal server di ServiceCenter.
<code>-imagepath:bitmap_path</code>	Specifica il percorso dei file immagine del client. Normalmente il programma di installazione Java copia i file immagine e li inserisce nella cache. Per ulteriori informazioni, vedere <i>Esecuzione di un client Java basato su browser</i> a pagina 35.

Aggiornamento automatico degli elenchi di record

Nella parte superiore della finestra suddivisa viene visualizzato un elenco di record. Il client Java può aggiornare gli elenchi di record in base a una pianificazione stabilita. Il client Java interroga il server e visualizza le eventuali modifiche nell'elenco di record.

Per impostare la frequenza di aggiornamento automatico per un elenco di record:

- 1 Dall'utilità Progettazione moduli aprire un modulo di elenco di record, ad esempio `probsummary.qbe.g`. Per ulteriori informazioni sull'utilità Progettazione moduli, consultare la *System Tailoring Guide, Volume 1* (in inglese).
- 2 Immettere il nome del modulo nel campo **Modulo**, ad esempio: `probsummary.qbe.g`.
- 3 Selezionare il record da aggiornare nell'utilità Progettazione moduli e premere **F9** oppure fare clic sul pulsante **Progetta**. Verrà visualizzata la modalità di progettazione moduli.

- 4 Fare clic nella tabella per visualizzare la finestra Proprietà.
- 5 Nella finestra Proprietà selezionare **RefreshRate**.
Immettere una frequenza di aggiornamento in secondi. Se si immette 0 la funzione sarà disattivata. Una frequenza di aggiornamento molto alta (inferiore a 15) può influire negativamente sull'interazione con il client Java.

Codici delle lingue

La versione 5.1 del client Java è stata localizzata in francese, tedesco e italiano; in ogni caso accetta dati in cinese (semplificato e tradizionale), giapponese, coreano, polacco, thailandese, turco e latino 1 ISO. La tabella seguente elenca i codici ISO a 2 caratteri per le lingue supportate dal client Java. Per ulteriori informazioni vedere *ServiceCenter Technical Reference* (in inglese).

Lingua	Identificatore
Chinese (cinese)	zh
French (francese)	fr
German (tedesco)	de
Italian (italiano)	it
Japanese (giapponese)	es
Korean (coreano)	ko
Polish (polacco)	pl
Thai (thailandese)	th
Turkish (turco)	tr

Esempio: Supporto per la lingua turca dell'applicazione autonoma

Per utilizzare il client Java come applicazione autonoma con la lingua turca, modificare il collegamento per l'avvio del client.

- 1 Individuare l'icona del collegamento al client Java sul desktop di Windows.
- 2 Fare clic con il pulsante destro del mouse e scegliere **Proprietà** dal menu di scelta rapida. Verrà visualizzata la finestra di dialogo Proprietà.
- 3 Fare clic sulla scheda **Collegamento**.
- 4 Nella casella di testo Destinazione aggiungere la seguente riga alla fine del comando:
`-codeset:tr -language:Turkish`
- 5 Fare clic su OK.

4 Server Hub

CAPITOLO

Il server hub consiste in un servlet Java che aggiunge un ulteriore livello di protezione al client Java, fungendo da proxy per il server di ServiceCenter. Invece di connettersi direttamente al server di ServiceCenter, il client Java si identifica presso il server hub, in genere tramite una connessione HTTP. Le connessioni dirette TCP/IP verso il client e il server di ServiceCenter sono realizzate attraverso il server hub, che provvede all'inoltro delle informazioni tra il client e il server.

Non è necessario che il client conosca l'ubicazione del server di ServiceCenter, pertanto il server di ServiceCenter può essere in esecuzione su un host diverso da quello del server hub. Il server hub può essere usato unitamente a un firewall e anche con più server di ServiceCenter. Una serie di parametri definiti sul lato client determinano se la connessione del client Java al server di ServiceCenter deve avvenire direttamente o indirettamente attraverso il server hub. I dettagli di connessione sono trasparenti.

Leggere questo capitolo per ottenere informazioni su

- *Requisiti del Server Hub* a pagina 80
- *Scenari di installazione* a pagina 83
- *Connessione callback* a pagina 84
- *Connessione diretta* a pagina 93
- *Connessione diretta senza server HTTP* a pagina 96
- *Configurazioni con firewall* a pagina 99
- *Supporto SSL del client Java* a pagina 102

Requisiti del Server Hub

Per l'installazione e l'esecuzione del server hub sono richiesti quattro componenti:

- Un server Web, facoltativo se si esegue un motore servlet autonomo in grado di inizializzare servlet senza connettersi al server HTTP.
- Motore servlet
- Client Java
- Server di ServiceCenter

Durante l'installazione standard del client Java viene installato anche il file del server hub. Consiste di un unico file JAR Java: `serverhb.jar`. Il percorso predefinito di questo file è `\\Peregrine\ServiceCenter\java\serverhb.jar`. La configurazione della connessione client Java/server hub prevede tre passaggi:

- Configurare il server Web per le zone servlet.
- Configurare il motore servlet in modo che includa il file `serverhb.jar`.
- Configurare il client Java di ServiceCenter per il collegamento al server hub.

Il pacchetto servlet server hub passato al motore servlet è `com.peregrine.hub.HubServlet`. Per il nome di questo pacchetto viene fatta distinzione tra maiuscole e minuscole.

Utilizzo di servlet Java

Il server hub è un servlet Java. I servlet Java sono programmi Java sul lato server in grado di generare contenuti dinamici. Supportano un modello richiesta/risposta comunemente utilizzato dai server. Sono più efficienti dei programmi CGI che avviano un nuovo processo ogni volta che vengono richiamati. Dal lato client, l'accesso a un servlet Java è analogo a quello di uno script CGI standard. I servlet Java sono indipendenti dalla piattaforma, nonché più veloci e sicuri degli script CGI.

Requisiti per il supporto di servlet

Il server hub richiede un server HTTP in grado di supportare servlet Java oppure un motore servlet autonomo in grado di supportare zone servlet senza ricorrere ai server HTTP. Sono disponibili estensioni servlet Java per i principali server Web per diverse piattaforme. Vi sono anche diversi server autonomi che offrono il supporto per i servlet Java. Maggiori informazioni su servlet Java, server e piattaforme sono disponibili sul sito Sun all'indirizzo <http://java.sun.com>.

Parametri del Server Hub

La seguente tabella elenca i parametri di connessione per il server hub relativi ai servlet del server hub. Questi parametri possono essere impostati durante la configurazione del motore servlet.

Parametro	Descrizione
servers	<p>Obbligatorio. Specifica un elenco, delimitato da punto e virgola, di uno o più server di ServiceCenter ai quali i client possono connettersi attraverso il server hub. Specificare ciascun server utilizzando la sintassi host:porta, con un alias o nome alternativo facoltativo utilizzabile dai client per identificare il server. Utilizzando un alias per ciascun server, il percorso di rete del server è trasparente per il client. Per impostare questo parametro utilizzando i due tipi di specifica del server:</p> <pre>servers=server1:12670;mainserver(server2:12670)</pre> <p>Il client deve identificare il primo server come server1:12670 e il secondo come mainserver.</p>
default	<p>Facoltativo. Specifica il nome (dagli elenchi dei server) del server di ServiceCenter predefinito utilizzato se il client non specifica alcun server. Ad esempio:</p> <pre>default=mainserver</pre>

Parametro	Descrizione
threads	Facoltativo. Specifica il numero massimo di thread allocati al server hub per la gestione delle connessioni dei client. Questo valore specifica il numero massimo di utenti simultanei supportati dal server hub. Se questo parametro viene omesso, verrà utilizzato il valore predefinito di 500. Ad esempio: threads=500
connectport	Obbligatorio. Specifica il numero di porta del server sul quale è in esecuzione il server hub utilizzabile dal client per connettersi direttamente al server hub, escludendo il meccanismo di callback standard. Questo parametro è facoltativo. Se non viene specificata alcuna porta, la funzione di connessione diretta è disabilitata. Per ulteriori informazioni, vedere <i>Parametri del file HTML</i> a pagina 82. Ad esempio: connectport=4331

Parametri del file HTML

La seguente tabella elenca i parametri di connessione che possono essere inseriti nel file HTML utilizzato per avviare il client Java (`scjava.htm`, `scjavamac.htm` oppure `scjavaplugin.htm`). Questi parametri si applicano anche al client Java autonomo. Impostare i parametri necessari secondo la propria configurazione.

La sintassi del parametro HTML è la seguente:

```
<param name="immettere un parametro" value="fornire il valore da applicare">
```

Parametro	Descrizione
HubURL	Obbligatorio. Specifica l'URL del server hub. Ad esempio: http://vostro_server_web:8080/hub/servlet/HubServlet Se impostato, il parametro HubURL ha priorità sui parametri Host e Service del client Java. Questo parametro può specificare anche l'ID server e la porta del server hub per il tipo di connessione diretta. Ad esempio: servername:12345 Se il parametro HubURL specifica l'URL del server hub, il nome dell'host e la porta per la connessione diretta verranno richiesti al server hub.

Parametro	Descrizione
HubAdapter	Facoltativo. Specifica il tipo di connessione con il server: SCEXpress oppure SCEXpressSL. Per aumentare la velocità, SCEXpressSL non restituisce alcuni codici di stato. Se la rete ha un tempo di risposta lento, specificare SCEXpressSL. Il valore predefinito è SCEXpress.
Server	Obbligatorio. Specifica il nome del server o l'alias del server hub per il server di ServiceCenter specificato nel parametro servers del server hub. Ad esempio: mainserver. Questo parametro è facoltativo solo se è stato impostato il parametro default del server hub.
ClientPort	Facoltativo. Specifica il numero di porta sul computer client utilizzabile dal client per la connessione al server hub. La porta deve essere disponibile per il client.
ConnectionType	Facoltativo. Specifica il tipo di connessione tra il server hub e il client. È possibile specificare callback o direct. Il valore predefinito è callback. La connessione diretta consente al client di avviare la connessione al server hub anche quando il client è protetto da un firewall oppure è in esecuzione su un computer che non ha un indirizzo IP statico. In una connessione callback, il server hub assegna al client Java una porta di collegamento in modo casuale. L'utilizzo di questa connessione presuppone che il client disponga di un indirizzo IP statico.

Scenari di installazione

Le modalità di installazione e configurazione del server hub dipendono dal modo in cui il client accede al server hub stesso. Ad esempio, il server hub consente ai client posti esternamente a un firewall di connettersi a un server di ServiceCenter situato all'interno di un firewall. Se non vi è un firewall tra il client e il server di ServiceCenter, non è necessario un server hub per stabilire la connessione tra il client Java e il server di ServiceCenter. Nell'installazione e configurazione del server hub, occorre innanzitutto considerare se il client utilizza un indirizzo IP statico o una connessione NAT (Network Address Translation).

Possono essere installati e configurati tre tipi diversi di connessioni con il server hub:

- Utilizzare una connessione callback quando il client Java è installato su una workstation con un indirizzo IP statico. Vedere *Connessione callback*.
- Utilizzare una connessione diretta quando il client Java viene avviato da una workstation con un indirizzo IP dinamico che utilizza NAT. Vedere *Connessione diretta* a pagina 93.
- Utilizzare una connessione diretta senza un server HTTP quando il motore servlet non richiede un server HTTP. Vedere *Connessione diretta senza server HTTP* a pagina 96.

Ciascun tipo di connessione con il server hub prevede tre passaggi. Per prima cosa deve essere configurato il motore servlet. Quindi è necessario configurare il client Java. Da ultimo, può essere necessario configurare il client Java per consentirne il collegamento attraverso uno o più firewall.

Connessione callback

Si tratta di una connessione molto semplice, utilizzata quando il client Java è installato su una workstation con un indirizzo IP statico. La sequenza di eventi è la seguente:

- Il client Java comunica al server hub il proprio indirizzo IP e la propria porta utilizzando una connessione HTTP.
- Il server hub apre una connessione TCP/IP di rimando all'indirizzo specificato.
- Il server hub apre una connessione TCP al server di ServiceCenter.
- Il server hub agisce in modo trasparente come server proxy.

La Figura 4-1 mostra la connessione callback dal client Java al server hub.

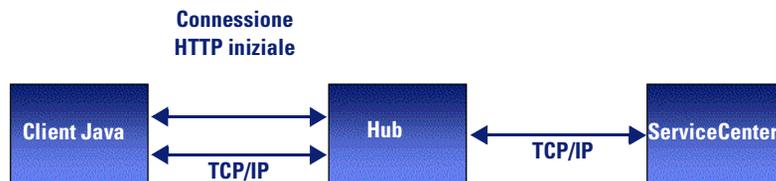


Figura 4-1: Connessione diretta

Passaggio 1: Configurazione del motore servlet

Il primo passaggio consiste nella configurazione del motore servlet per l'interazione con il server hub. Il primo esempio utilizza Tomcat. Il secondo esempio, che inizia alla pagina 88, utilizza JRun.

Esempio con Tomcat

Apache Tomcat è un motore servlet open source disponibile per la maggior parte delle piattaforme. Il client Java richiede Tomcat 4.1 o successivo. Per maggiori informazioni sul server Web Apache e sul motore servlet Tomcat, fare riferimento a www.apache.org. Il motore servlet Tomcat non richiede un server Web per gestire i servlet se viene configurato come motore servlet autonomo.

Installare e configurare Tomcat come indicato nella corrispondente documentazione del prodotto. Nel seguente esempio, la directory di installazione di Tomcat è TOMCAT_HOME.

Per configurare il motore servlet Tomcat:

- 1 Per implementare il server hub Peregrine, è necessario modificare la struttura delle directory di Tomcat. Sotto `TOMCAT_HOME/webapps`, aggiungere una nuova directory denominata `hub` per il servlet hub. Sotto la directory `hub`, aggiungere una sottodirectory denominata `WEB-INF` e una sottodirectory di `Web-inf` denominata `lib`. La Figura 4-2 mostra le modifiche apportate alla struttura delle directory di Tomcat.

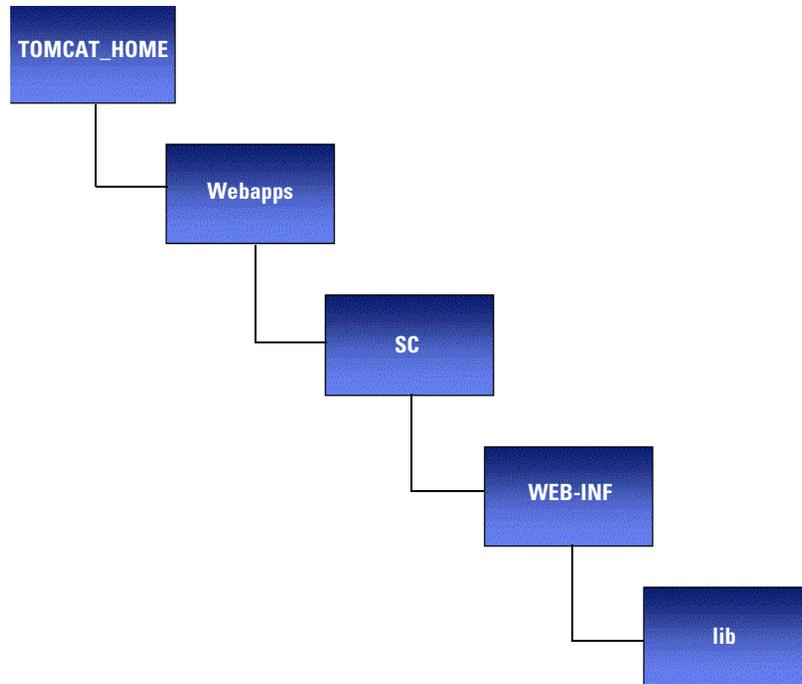


Figura 4-2: Configurazione della directory Tomcat

- 2 Utilizzando un editor di testo, creare un file denominato `web.xml` e salvarlo nella directory `Web-inf`. Il file `web.xml` specifica i parametri del server hub. Il seguente è un esempio di file `web.xml` con i parametri `servers` e `default` impostati. Per ulteriori informazioni, vedere *Parametri del Server Hub* a pagina 81.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <servlet>
    <servlet-name>Hub</servlet-name>
    <servlet-class>com.peregrine.hub.HubServlet</servlet-class>
    <init-param>
      <param-name>servers</param-name>
      <param-value>mainserver(vostro_server_servicecenter:12670)
      </param-value>
    </init-param>
    <init-param>
      <param-name>default</param-name>
      <param-value>mainserver</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Hub</servlet-name>
    <url-pattern>*</url-pattern>
  </servlet-mapping>
</web-app>

```

- 3 Copiare il file \\...\java\serverhb.jar dalla directory di installazione della workstation alla directory lib.
- 4 Riavviare Tomcat.
- 5 Se sono state osservate le convenzioni per l'assegnazione del nome alle directory specificate nell'esempio, eseguire il test della configurazione collegandosi a:

http://vostro_server_web:numerporta/hub/servlet/Hub?cmd=1

Se il server hub funziona correttamente, verrà visualizzata una pagina vuota o una finestra di dialogo che informa che la pagina non contiene dati. In caso contrario, rivedere tutte le impostazioni di configurazione e verificare che il percorso delle classi sia stato configurato correttamente. Il parametro da inserire nel file HTML usato per avviare il client Java è:

```

<param name="HubURL"
value="http://vostro_server_web:numerporta/hub/servlet/HubServlet/">

```

Note

- 1 Per il collegamento al server hub è necessario specificare il valore del parametro **HubURL** nel file HTML che avvia il client Java. Per ulteriori informazioni, vedere *Passaggio 2: Configurazione del client Java* a pagina 90.
- 2 Se Tomcat viene eseguito come applicazione autonoma, è possibile specificare l'URL del server hub senza passare alcun numero di porta. Ad esempio, è possibile utilizzare un URL come `http://localhost/servlet/hub` invece di `http://localhost:8080/servlet/hub`.

Esempio con JRun 3.1

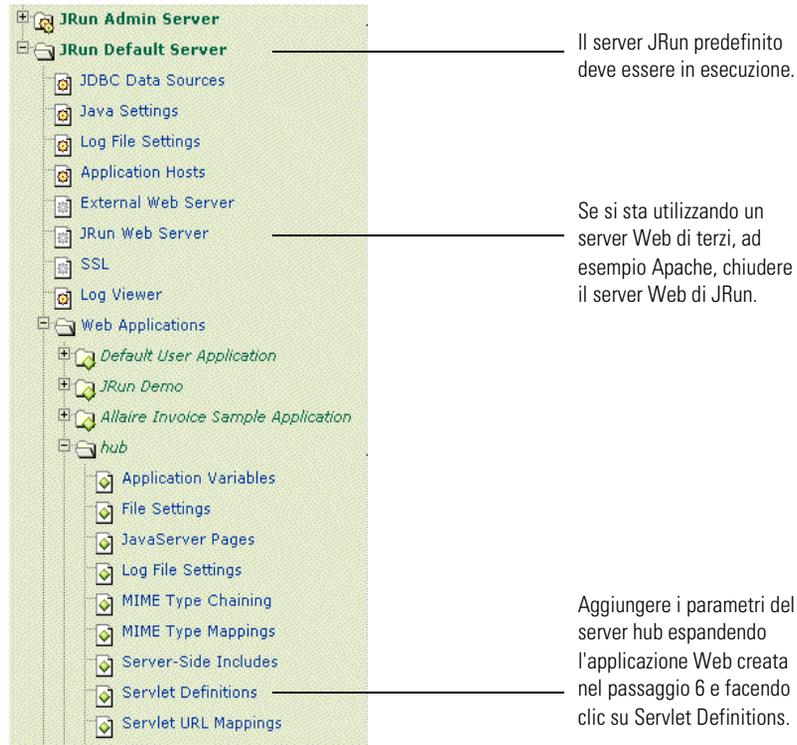
JRun è un motore commerciale che installa, per impostazione predefinita, due server Web. È necessario configurare JRun in modo che riconosca un server Web di altri produttori, come Apache o IIS. Per maggiori informazioni su JRun, fare riferimento a www.macromedia.com.

Per configurare un motore servlet JRun:

- 1 Installare e configurare JRun facendo riferimento alla specifica documentazione.
- 2 Avviare lo strumento di amministrazione di JRun.
- 3 Aprire il ramo **JRun Default Server** come mostrato nella Figura 4-3 a pagina 89.
- 4 Fare clic su **Java Settings**. Nel riquadro di destra verranno visualizzate le impostazioni relative a Java.
- 5 Fare clic su **Classpath** e aggiungere il percorso al file `serverhb.jar`.
- 6 Fare clic su **Update**.
- 7 Dopo aver aggiunto il file `serverhb.jar` al percorso delle classi JRun, è necessario specificare in JRun come avviare il server hub. Fare clic su **Web Applications** nella struttura dei menu dello strumento di amministrazione di JRun. Nel riquadro di destra sarà visualizzata la procedura guidata **Deploy Web Applications**. Utilizzare la procedura guidata per creare un'applicazione Web per il server hub.

Nota: All'interno della procedura guidata definire una mappatura URL per il server hub. La mappatura informa il server Web che le richieste per determinati URL, ad esempio `/hub`, devono essere gestite da JRun e non dal server Web.

- 8 L'applicazione Web creata al passaggio 7 a pagina 88 sarà vuota. Per definire i parametri del server hub, espandere la struttura delle Web Applications e selezionare l'applicazione Web creata al passaggio 7, come illustrato nella Figura 4-3.



Il server JRun predefinito deve essere in esecuzione.

Se si sta utilizzando un server Web di terzi, ad esempio Apache, chiudere il server Web di JRun.

Aggiungere i parametri del server hub espandendo l'applicazione Web creata nel passaggio 6 e facendo clic su Servlet Definitions.

Figura 4-3: Struttura di configurazione di JRun

- 9 Fare clic su **Servlet Definitions**.
- 10 Fare clic su **Edit** nel riquadro di destra. Verrà visualizzata la finestra di modifica di JRun.

- 11 Nella finestra di modifica di JRun, immettere i seguenti valori:
 - Nel campo **Name** digitare **hub**. Nel campo **Class Name** digitare: `com.peregrine.hub.HubServlet`
 - Scorrere la finestra di modifica di Jrun verso destra per visualizzare il campo **Init Arguments**. Nel campo **Init Arguments** digitare i parametri di inizializzazione del server hub. Ad esempio:


```
servers=mainserver(vostro_server_servicecenter:12670)
threads=500
default=mainserver
```

Per una descrizione completa dei parametri e dei valori di inizializzazione vedere *Parametri del Server Hub* a pagina 81.

- 12 Riavviare il server Web.
- 13 Se sono state osservate le convenzioni per l'assegnazione del nome alle directory specificate nell'esempio, eseguire il test della configurazione collegandosi a:


```
http://www.vostroserver.com/hub/servlet/HubServlet?cmd=1
```

Se il server hub funziona correttamente, verrà visualizzata una pagina vuota o una finestra di dialogo che informa che la pagina non contiene dati. In caso contrario, rivedere tutte le impostazioni di configurazione e verificare che il percorso delle classi sia stato configurato correttamente.

Passaggio 2: Configurazione del client Java

Per avviare il client Java possono essere utilizzati due metodi. Se si sceglie di utilizzare il metodo 2, specificare la porta di connessione nel file HTML usato per avviare il client Java. I file HTML sono `scjava.htm`, `scjavamac.htm` oppure `scjavaplugin.htm`.

- Metodo 1: Il client Java seleziona una porta di connessione al server hub in modo casuale. Scegliere questo metodo se non è presente alcun firewall tra client e server.
- Metodo 2: Specificare la porta di connessione nel file HTML utilizzato per avviare il client Java. Scegliere questo metodo se tra client e server è presente un firewall. La porta di connessione definita nel file HTML deve essere aperta al traffico in ingresso. Per maggiori informazioni sulla porta corretta, rivolgersi all'amministratore di sistema o Web locale.

Il client Java deve riconoscere il server hub per poter stabilire una connessione con l'hub. Le seguenti impostazioni devono essere inserite nel file HTML usato per avviare il client Java, oppure passate nella riga di comando per l'applicazione autonoma.

Parametri metodo 1

Utilizzare i parametri elencati nella seguente tabella quando il client Java non è protetto da un firewall e la porta di connessione può essere selezionata in modo casuale.

Parametro	Definizione
HubURL	<p>Specifica l'URL del server hub. Ad esempio:</p> <pre><paramname="HubURL" value= http://vostro_server_web:num_porta/hub/servlet/HubServlet></pre> <p>Se impostato, il parametro HubURL ha la priorità sui parametri host e service del client Java.</p>
HubAdapter	<p>Specifica il tipo di connessione da instaurare con il server. Specificare SCEXpress.</p>
Server	<p>Specifica il nome del server o l'alias del server hub per il server di ServiceCenter a cui ci si deve collegare e che è stato definito nel parametro servers del server hub (ad esempio, mainserver). Questo parametro è facoltativo se è stato impostato il parametro default del server hub.</p>

Esempio di file HTML

Nel file HTML utilizzato per avviare il client Java, sostituire i parametri Host e Service con i seguenti parametri.

```
<param name="HubURL" value="http://www.host.com:8001/servlet/hub">
<param name="Server" value="mainserver">
<param name="HubAdapter" value="SCEXpress">
```

dove è necessario sostituire `www.host.com:8001/servlet/hub` con l'URL del servlet del server hub e `mainserver` con il nome o l'alias del server di ServiceCenter.

Esempio relativo alla applicazione autonoma

Quando il client Java viene eseguito in modalità autonoma, i parametri possono essere passati al client sotto forma di argomenti della riga di comando. Quindi, per connettersi al server di ServiceCenter attraverso il server hub si sostituiranno i parametri **Host** e **Service** con quanto segue:

```
-HubURL=http://www.host.com:8001/servlet/hub -Server=mainserver
-HubAdapter=SCExpress
```

dove è necessario sostituire **www.host.com:8001/servlet/hub** con l'URL del servlet del server hub e **mainserver** con il nome o l'alias del server di ServiceCenter. I parametri della modalità autonoma possono essere specificati anche nel file `scj.ini`.

Parametri metodo 2

Quando il client Java è protetto da un firewall, utilizzare i parametri elencati nella seguente tabella. La porta specificata nel file HTML utilizzato per avviare il client Java deve essere aperta al traffico in ingresso. Per maggiori informazioni, rivolgersi al proprio amministratore di sistema o Web.

Parametro	Definizione
HubURL	<p>Specifica l'URL del server hub. Ad esempio:</p> <pre><paramname="HubURL" value="http://vostro_server_web:num_porta/hub/servlet/HubServlet"></pre> <p>Se impostato, il parametro HubURL ha la priorità sui parametri host e service del client Java.</p>
HubAdapter	<p>Specifica il tipo di connessione da instaurare con il server. Specificare SCExpress.</p>
ClientPort	<p>Specifica il numero di porta sul computer client utilizzabile dal client per la connessione al server hub. La porta deve essere disponibile per il client.</p>
Server	<p>Specifica il nome del server o l'alias del server hub per il server di ServiceCenter a cui ci si deve collegare e che è stato definito nel parametro servers del server hub (ad esempio, mainserver). Questo parametro è facoltativo se è stato impostato il parametro default del server hub.</p>

Esempio di file HTML

Nel file HTML utilizzato per avviare il client Java, sostituire i parametri Host e Service con i seguenti parametri.

```
<param name="HubURL" value="http://www.host.com:8001/servlet/hub">
<param name="Server" value="mainserver">
<param name="HubAdapter" value="SCExpress">
```

dove è necessario sostituire `www.host.com:8001/servlet/hub` con l'URL del servlet del server hub e `mainserver` con il nome o l'alias del server di ServiceCenter.

Esempio relativo alla applicazione autonoma

Quando il client Java viene eseguito in modalità autonoma, i parametri possono essere passati al client sotto forma di argomenti della riga di comando. Quindi, per connettersi al server di ServiceCenter attraverso il server hub si sostituiranno i parametri Host e Service con quanto segue:

```
-HubURL=http://www.host.com:8001/servlet/hub -Server=mainserver
-HubAdapter=SCExpress -ClientPort=12345
```

dove è necessario sostituire `www.host.com:8001/servlet/hub` con l'URL del servlet del server hub e `mainserver` con il nome o l'alias del server di ServiceCenter. I parametri della modalità autonoma possono essere specificati anche nel file `scj.ini`.

Connessione diretta

Il client Java apre una connessione diretta con il server hub attraverso una porta specificata. Se si sceglie questo tipo di connessione, il server Web non assegna al client Java una porta casuale e la rete non assegna un indirizzo IP dinamico. L'amministratore di sistema dovrà assegnare al client Java anche una porta sul firewall. La sequenza di eventi è la seguente:

- Il client Java invia al server hub una richiesta di collegamento utilizzando una connessione HTTP.
- Il server hub restituisce un indirizzo IP e un numero di porta utilizzando la stessa connessione HTTP.
- Il client apre una connessione TCP/IP con il server hub.
- Il server hub è trasparente come server proxy.

La Figura 4-4 mostra la connessione a un server di ServiceCenter mediante un server hub. Impostare il parametro `ConnectionType` nel file HTML utilizzato per avviare il client Java. Impostare il parametro `connectport` nel motore servlet che fa da contenitore per il server hub.

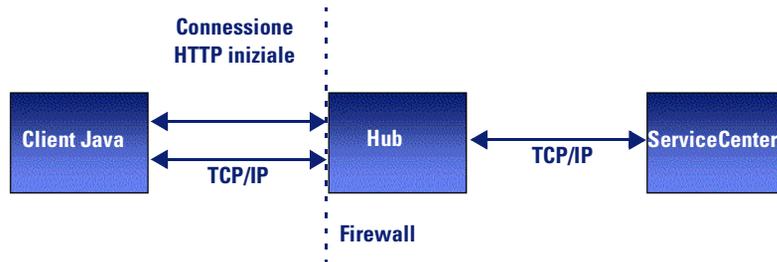


Figura 4-4: Connessione diretta

Passaggio 1: Configurazione del motore servlet

Il primo passo in questo scenario consiste nella configurazione del motore servlet per l'interazione con il server hub.

- 1 Configurare il motore servlet come illustrato al *Passaggio 1: Configurazione del motore servlet* a pagina 85 e procedere con il passaggio seguente.
- 2 Se si utilizza un server Tomcat, aggiungere le seguenti righe al file `web.xml`:

```
<init-param>
  <param-name>connectport</param-name>
  <param-value>numero_porta</param-value>
</init-param>
```

Avviso: Verificare che la porta specificata sia disponibile per le connessioni in ingresso.

Passaggio 2: Configurazione del client Java

Il client Java deve riconoscere il server hub per potervi stabilire una connessione. Utilizzare i parametri della seguente tabella nel file HTML utilizzato per avviare il client Java. Se l'applicazione è di tipo autonomo, i parametri possono essere passati anche sulla riga di comando.

Parametro	Definizione
HubURL	Specifica l'URL del server hub. Ad esempio: http://www.host.com:8001/servlet/hub
HubAdapter	Facoltativo. Specifica il tipo di connessione da instaurare con il server. È possibile specificare SCEXpress.
ConnectionType	Specifica il tipo di connessione tra il server hub e il client. È possibile specificare callback o direct. Scegliere direct. La connessione diretta consente al client di avviare la connessione al server hub anche quando il client è protetto da un firewall oppure è in esecuzione su un computer che non ha un indirizzo IP statico.
Server	Specifica il nome del server o l'alias del server hub per il server di ServiceCenter a cui ci si deve collegare e che è stato definito nel parametro servers del server hub (ad esempio, mainserver). Questo parametro è facoltativo se è stato impostato il parametro default del server hub.

Nota: Se il parametro **HubURL** è impostato, avrà la precedenza sui parametri **Host** e **Service** del client Java. Per ragioni di protezione, non impostare i parametri **Host** e **Service** quando si utilizza un server hub.

Esempio di file HTML

Il client Java può essere eseguito come applet e connettersi al server di ServiceCenter mediante il server hub. Nel file HTML utilizzato per avviare il client Java, sostituire i parametri **Host** e **Service** con i seguenti parametri.

```
<param name="HubURL" value="http://www.host.com:8001/servlet/hub">
<param name="Server" value="mainserver">
<param name="ConnectionType" value="direct">
<param name="HubAdapter" value="SCEXpress">
```

dove è necessario sostituire **www.host.com:8001/servlet/hub** con l'URL del servlet del server hub e **mainserver** con il nome o l'alias del server di ServiceCenter.

Esempio relativo alla applicazione autonoma

Quando il client Java viene eseguito in modalità autonoma, i parametri possono essere passati al client sotto forma di argomenti della riga di comando:

```
-HubURL=http://www.host.com:8001/servlet/hub -Server=mainserver  
-HubAdapter=SCExpress -ConnectionType=direct
```

dove è necessario sostituire `www.host.com:8001/servlet/hub` con l'URL del servlet del server hub e `mainserver` con il nome o l'alias del server di ServiceCenter. I parametri della modalità autonoma possono essere specificati anche nel file `scj.ini`.

Connessione diretta senza server HTTP

È possibile evitare del tutto la connessione HTTP specificando un server mediante il parametro `HubURL` nel file HTML usato per avviare il client Java. Ad esempio, si può specificare `servername:12345` invece di `http://servername/hub`. Il motore servlet utilizzato deve supportare l'avvio di zone servlet senza ricorrere a un server HTTP. Per ulteriori informazioni, vedere la documentazione del motore servlet. L'omissione del server HTTP può migliorare il grado di protezione del sistema.

Il client Java apre una connessione diretta con il server hub attraverso una porta specificata. Se si utilizza questo metodo, l'amministratore del sistema deve rendere disponibile al client Java una porta attraverso il firewall. La sequenza di eventi è la seguente:

- Il client Java invia al server hub una richiesta di collegamento utilizzando una connessione TCP/IP.
- Il server hub restituisce il proprio indirizzo IP e numero di porta utilizzando la stessa connessione TCP/IP.
- Il client apre una connessione TCP/IP con il server hub.
- Il server hub è trasparente come server proxy.

La Figura 4-5 mostra una connessione diretta senza server HTTP. Il motore servlet utilizzato deve supportare le zone servlet senza necessità di un server HTTP. Impostare il parametro **ConnectionType** nel file HTML utilizzato per collegarsi al server hub. Impostare il parametro **connectport** nel motore servlet che fa da contenitore per il server hub.

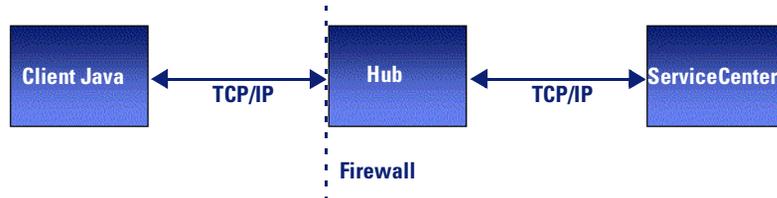


Figura 4-5: Connessione diretta senza server HTTP

Passaggio 1: Configurazione del motore servlet

Per configurare il motore servlet, eseguire i passaggi indicati in *Connessione diretta* a pagina 93.

Passaggio 2: Configurazione del client Java

Il client Java deve riconoscere il server hub per potervi stabilire una connessione. Utilizzare i parametri della seguente tabella nel file HTML utilizzato per avviare il client Java. Se l'applicazione è di tipo autonomo, i parametri possono essere passati anche sulla riga di comando.

Parametro	Definizione
HubURL	Specifica l'URL del server hub. Ad esempio: servername:12345 Se si imposta il parametro HubURL con l'URL del server hub, il nome e la porta dell'host verranno richiesti al server hub.
HubAdapter	Specifica il tipo di connessione da instaurare con il server. Specificare SCEXpress.

Parametro	Definizione
ConnectionType	<p>Specifica il tipo di connessione tra il server hub e il client. È possibile specificare callback o direct. Scegliere direct.</p> <p>La connessione diretta consente al client di avviare la connessione al server hub anche quando il client è protetto da un firewall oppure è in esecuzione su un computer che non ha un indirizzo IP statico.</p>
Server	<p>Specifica il nome del server o l'alias del server hub per il server di ServiceCenter a cui ci si deve collegare e che è stato definito nel parametro servers del server hub (ad esempio, mainserver). Questo parametro è facoltativo se è stato impostato il parametro default del server hub.</p>

Nota: Se il parametro **HubURL** è impostato, avrà la precedenza sui parametri **Host** e **Service** del client Java. Per ragioni di protezione, non impostare i parametri **Host** e **Service** quando si utilizza un server hub.

Esempio di file HTML

Il client Java può essere eseguito come applet e connettersi al server di ServiceCenter mediante il server hub. Nel file HTML utilizzato per avviare il client Java, sostituire i parametri **Host** e **Service** con i seguenti parametri.

```
<param name="HubURL" value="servername:12345">
<param name="Server" value="mainserver">
<param name="ConnectionType" value="direct">
<param name="HubAdapter" value="SCEExpress">
```

dove è necessario sostituire **servername:12345** con il nome host e la porta della connessione diretta del server hub e **mainserver** con il nome o l'alias del server di ServiceCenter.

Esempio relativo alla applicazione autonoma

Quando il client Java viene eseguito in modalità autonoma, i parametri possono essere passati al client sotto forma di argomenti della riga di comando. Quindi, per connettersi al server di ServiceCenter attraverso il server hub si sostituiranno i parametri **Host** e **Service** con quanto segue:

```
-HubURL=servername:12345 -Server=mainserver
-HubAdapter=SCEExpress -ConnectionType=direct
```

dove è necessario sostituire **servername:12345** con il nome host e la porta della connessione diretta del server hub e **mainserver** con il nome o l'alias del server di ServiceCenter. I parametri della modalità autonoma possono essere specificati anche nel file **scj.ini**.

Configurazioni con firewall

Se si prevede di proteggere il server hub con un firewall sul lato client o sul lato server (o su entrambi), è necessario configurare il firewall correttamente. Può essere necessario modificare le configurazioni descritte in questa sezione per adattarle al proprio ambiente di lavoro. La Figura 4-6 a pagina 99 mostra una tipica configurazione con un firewall sui due lati del server hub. Il firewall tra il client Java e il server hub deve essere aperto al traffico in ingresso per consentire il collegamento del client al server hub. Il firewall tra il server hub e ServiceCenter deve avere una porta aperta al traffico ServiceCenter in ingresso verso l'indirizzo IP usato dal server hub.

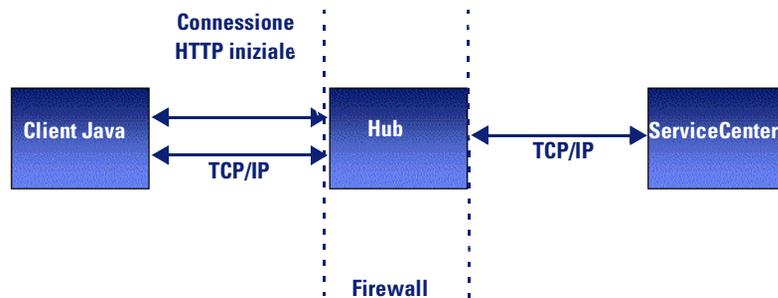


Figura 4-6: Configurazioni con firewall

Client Java protetto da firewall

Se il client Java è protetto da un firewall, occorre configurare il client Java e il server hub in uno di due modi, a seconda della modalità di assegnazione dell'indirizzo IP del client.

Indirizzo IP	Configurazione del client Java
Indirizzo IP statico	Vedere <i>Connessione callback</i> a pagina 84.
NAT (Network Address Translation)	Vedere <i>Connessione diretta</i> a pagina 93.

Server hub protetto da firewall

Se il server hub è protetto da un firewall, occorre configurare il client Java in uno di due modi, a seconda della modalità di assegnazione dell'indirizzo IP del client.

Indirizzo IP	Configurazione del client Java
Client Java con indirizzo IP statico	<p>In questa configurazione, il client Java può instaurare una connessione callback con il server hub se sono soddisfatte le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il parametro ClientPort per il client Java è stato specificato nel file HTML utilizzato per avviare il client Java. • Il firewall deve essere aperto per consentire la connessione del client Java sulla porta specificata mediante il parametro ClientPort. Per ulteriori informazioni, vedere <i>Connessione callback</i> a pagina 84.
Client Java con NAT	<p>In questa configurazione, il client Java può instaurare una connessione diretta con il server hub se sono soddisfatte le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il server hub è inizializzato con il parametro connectport • Il firewall deve essere configurato per consentire al server hub di accettare connessioni in ingresso sulla porta specificata dal parametro connectport. Per ulteriori informazioni, vedere <i>Connessione diretta</i> a pagina 93.

Esempio **Procedere come segue:**

- 1 Configurare il motore servlet eseguendo le operazioni indicate al *Passaggio 1: Configurazione del motore servlet* a pagina 85.
- 2 Se si utilizza un server Tomcat, aggiungere le seguenti righe al file `web.xml`:

```
<init-param>
  <param-name>connectport</param-name>
  <param-value>numero_porta</param-value>
</init-param>
```

Verificare che la porta specificata sia disponibile per le connessioni in ingresso.

- 3 Se si utilizza un server JRun, aggiungere il seguente parametro alla definizione dei servlet:

connectport=numero_porta_disponibile.

Specificare i seguenti parametri nel file HTML usato per avviare il client Java:

Parametro	Definizione
HubURL	Specifica l'URL del server hub. Ad esempio: http://www.host.com:8001/servlet/hub
HubAdapter	Specifica il tipo di connessione da instaurare con il server. Specificare SCEXpress.

Esempio di file HTML

Nel file HTML utilizzato per avviare il client Java, sostituire i parametri **Host** e **Service** con i seguenti parametri.

```
<param name="HubURL" value="http://www.host.com:8001/servlet/hub">
<param name="Server" value="mainserver">
<param name="HubAdapter" value="SCEXpress">
```

dove è necessario sostituire **www.host.com:8001/servlet/hub** con l'URL del servlet del server hub e **mainserver** con il nome o l'alias del server di ServiceCenter.

Esempio relativo alla applicazione autonoma

Quando il client Java viene eseguito in modalità autonoma, i parametri possono essere passati al client sotto forma di argomenti della riga di comando. Quindi, per connettersi al server di ServiceCenter attraverso il server hub si sostituiranno i parametri **Host** e **Service** con quanto segue:

```
-HubURL=http://www.host.com:8001/servlet/hub -Server=mainserver
-HubAdapter=SCEXpress
```

dove è necessario sostituire **www.host.com:8001/servlet/hub** con l'URL del servlet del server hub e **mainserver** con il nome o l'alias del server di ServiceCenter. I parametri della modalità autonoma possono essere specificati anche nel file **scj.ini**.

Supporto SSL del client Java

Il protocollo SSL (Secure Socket Layer) offre funzioni di autenticazione, crittografia e protezione dell'integrità di dati riservati. È possibile abilitare il supporto del protocollo SSL nelle comunicazioni tra client Java e server hub. Il client Java è in grado di collegarsi al server hub mediante connessioni dirette o *callback*, tuttavia supporta il protocollo SSL solo nelle connessioni dirette. Per ulteriori informazioni, vedere *Connessione diretta* a pagina 93.

Il supporto SSL di ServiceCenter utilizza un certificato autofirmato per autenticare il server. Per l'autenticazione e lo scambio delle chiavi viene utilizzato per impostazione predefinita l'algoritmo crittografico RSA a 1024 bit. Questo valore è utilizzato solo per lo scambio delle chiavi. Il canale è codificato a 128 bit. È possibile specificare valori delle chiavi e certificati in un file `scssl.ini`.

Per informazioni aggiornate sulla compatibilità SSL del client Java di ServiceCenter per JRE specifici, vedere *Sito Web CenterPoint di Peregrine* a pagina 9.

Per maggiori informazioni su Java Secure Socket Extensions (JSSE), vedere la *Java Secure Socket Extension (JSSE) Reference Guide for the Java 2 SDK, Standard Edition, v 1.4 (in inglese)*. La guida è disponibile all'indirizzo <http://java.sun.com/j2se/1.4/docs/guide/security/jsse/JSSERefGuide.html>.

Requisiti di sistema per SSL

Il server hub richiede JRE versione 1.3.x e Apache Tomcat versione 4.1.x. Il client richiede JRE versione 1.3.x.

Abilitazione del supporto SSL

L'abilitazione del supporto SSL richiede alcune semplici operazioni.

Passaggio 1 Abilitare il supporto SSL sul server. Vedere *Supporto SSL lato server*.

Passaggio 2 Abilitare il supporto SSL sul client. Vedere *Supporto SSL lato client* a pagina 105.

Passaggio 3 Verificare che SSL sia stato abilitato correttamente. Vedere *Verifica SSL per server e client* a pagina 109.

Supporto SSL lato server

Informazioni preliminari

- 1 Rivedere le informazioni fornite nel presente capitolo riguardo alle relazioni tra client, server e server hub.
- 2 Installare il server hub utilizzando il metodo della connessione diretta. Vedere *Connessione diretta* a pagina 93 oppure *Connessione diretta senza server HTTP* a pagina 96.
- 3 Ottenere la chiave:
 - Utilizzare la chiave di crittografia predefinita fornita con il server hub e passare a *Utilizzo di una chiave di crittografia predefinita*.
 - Utilizzare una chiave di crittografia privata e passare a *Utilizzo di una chiave di crittografia privata* a pagina 103.

Utilizzo di una chiave di crittografia predefinita

Per abilitare SSL sul server, eseguire le seguenti operazioni:

- 1 Aprire questo file con un editor di testo:
TOMCAT_HOME>/webapps/<NOME_SERVER_HUB>/WEB-INF/web.xml
- 2 Individuare l'ultima istanza del tag </init-param>.
- 3 Nella riga seguente, inserire questo nuovo tag <init-param>:

```
<init-param>
  <param-name>sslconnectport</param-name>
  <param-value>portvalue</param-value>
</init-param>
```
- 4 Salvare e chiudere il file.
- 5 Seguire le istruzioni fornite in *Supporto SSL lato client* a pagina 105.

Utilizzo di una chiave di crittografia privata

Per abilitare SSL sul server, eseguire le seguenti operazioni:

- 1 Aprire questo file con un editor di testo:
TOMCAT_HOME/webapps/<NOME_SERVER_HUB>/WEB-INF/web.xml
- 2 Individuare l'ultima istanza del tag </init-param>.

- 3 Nella riga seguente, inserire questo nuovo tag `<init-param>`:

```
<init-param>
  <param-name>sslconnectport</param-name>
  <param-value>portvalue</param-value>
</init-param>
<init-param>
  <param-name>scjssl</param-name>
  <param-value>sslserver.ini</param-value>
</init-param>
```

- 4 Salvare e chiudere il file.

- 5 Creare un nuovo file con un editor di testo.

- 6 Digitare le seguenti due righe:

```
Keystore=peregrine.key
Keystorepass=password
```

- 7 Salvare il file come `sslserver.ini` nella seguente directory:

```
TOMCAT_HOME/webapps/<NOME_SERVER_HUB>/WEB-INF
```

- 8 Chiudere il file.

- 9 Creare un file di chiave contenente la chiave privata. Assegnare al file il nome `peregrine.key`.

Nota: Ogni JRE fornisce uno strumento di gestione delle chiavi di crittografia che crea un file con la chiave crittografata. Per ulteriori informazioni, vedere la documentazione JRE.

- 10 Se è in esecuzione JRE 1.4.x, l'installazione sul lato server è completa. Procedere con le istruzioni fornite in *Supporto SSL lato client* a pagina 105. Se si esegue JRE 1.3.x, eseguire le operazioni descritte in *Recupero delle librerie SSL (solo JRE 1.3.x)*.

Recupero delle librerie SSL (solo JRE 1.3.x)

Effettuare le seguenti operazioni per scaricare le librerie SSL:

- 1 Scaricare le librerie SSL di Sun (JSSE) da uno dei seguenti indirizzi:
 - <http://java.sun.com/products/jsse/>
 - <http://support.peregrine.com>
- 2 Copiare i seguenti file di libreria da quelli scaricati:
 - jsse.jar
 - jnet.jar
 - jcert.jar
- 3 Incollare questi tre file nella directory `../lib/ext directory` di JRE 1.3.
- 4 Arrestare e riavviare il server applicazioni.

Avviso: L'arresto e il riavvio del server applicazioni arresta tutte le sessioni utente del Client Java.

Supporto SSL lato client

Il server supporta due tipi di connessioni client, a seconda della chiave.

- Se il client si collega al server con la chiave di crittografia predefinita fornita con il server hub, vedere *Collegamento di un client Java Windows con una chiave predefinita* o *Collegamento di un client Java basato su browser con una chiave predefinita* a pagina 106.
- Se il client si collega al server con una chiave di crittografia privata, vedere *Collegamento di un client Java Windows con una chiave privata* a pagina 107 o *Collegamento di un client Java basato su browser con una chiave privata* a pagina 108.

Collegamento di un client Java Windows con una chiave predefinita

Procedere come segue:

- 1 Dal menu Start di Windows, scegliere Programmi > Peregrine ServiceCenter.
- 2 Fare clic con il pulsante destro del mouse su Client Java, quindi scegliere Proprietà.
- 3 Nella scheda Collegamento, aggiungere quanto segue al termine del contenuto del campo Destinazione:

-sslenabled=true

Ad esempio:

```
C:\jdk1.3.1_07\jre\bin\java.exe -classpath jbird.jar
com.peregrine.sc.client.ClientApplication
-HubURL=http://<servername>:8080/hub/hub -Server=mainserver
-ConnectionType=direct -sslenabled=true
```

- 4 Fare clic su Applica.

Windows NT:

Non è possibile eseguire il passaggio 2. Creare un collegamento per il client Java sul desktop e fare clic con il pulsante destro del mouse per modificare il campo di destinazione.

Collegamento di un client Java basato su browser con una chiave predefinita

Procedere come segue:

- 1 Dal menu Start di Windows, scegliere Programmi > Peregrine ServiceCenter.
- 2 Fare clic con il pulsante destro del mouse sul client Java basato su browser e scegliere Proprietà.
- 3 Nella scheda Collegamento, individuare il campo Destinazione.
- 4 Aprire il file //Peregrine/ServiceCenter/java/scjava.htm con un editor di testo.
- 5 Aggiungere il seguente comando:
<param name= "sslenabled" value="true">

- 6 Salvare e chiudere il file.

Windows NT:

Non è possibile eseguire il passaggio 2 a pagina 106. Creare un collegamento per il client Java sul desktop e fare clic con il pulsante destro del mouse per modificare il campo di destinazione.

Nota: Per ulteriori informazioni, vedere *File .htm del client Java* a pagina 60.

Collegamento di un client Java Windows con una chiave privata

- 1 Dal menu Start di Windows, scegliere Programmi > Peregrine ServiceCenter.
- 2 Fare clic con il pulsante destro del mouse su Client Java, quindi scegliere Proprietà.
- 3 Nella scheda Collegamento, aggiungere quanto segue al termine del contenuto del campo Destinazione:

```
-sslenabled=true -scjssl=sslclient.ini
```

Ad esempio:

```
C:\jdk1.3.1_07\jre\bin\java.exe -classpath jbird.jar
com.peregrine.sc.client.ClientApplication
-HubURL=http://<servername>:8080/hub/hub -Server=mainserver
-ConnectionType=direct -sslenabled=true -scjssl=sslclient.ini
```

- 4 Fare clic su Applica.

Windows NT:

Non è possibile eseguire il passaggio 2. Creare un collegamento per il client Java sul desktop e fare clic con il pulsante destro del mouse per modificare il campo di destinazione.

Collegamento di un client Java basato su browser con una chiave privata

- 1 Dal menu **Start** di Windows, scegliere **Programmi > Peregrine ServiceCenter**.
- 2 Fare clic con il pulsante destro del mouse sul client **Java** basato su browser e scegliere **Proprietà**.
- 3 Nella scheda **Collegamento**, individuare il campo **Destinazione**.

Windows NT:

Non è possibile fare clic con il pulsante destro del mouse sul client Java. Creare un collegamento per il client Java sul desktop e fare clic con il pulsante destro del mouse per modificare il campo di destinazione.

- 1 Aprire il file `//Peregrine/ServiceCenter/java/scjava.htm` con un editor di testo.
- 2 Aggiungere il seguente comando:


```
<param name= "sslenabled" value="true">
<param name= "scjssl" value="sslclient.ini">
```
- 3 Salvare e chiudere il file.

Nota: Per ulteriori informazioni, vedere *File .htm del client Java* a pagina 60.

- 4 Eseguire le operazioni descritte nella sezione *Creazione di un file truststore*.

Creazione di un file truststore

L'ambiente JRE offre uno strumento di gestione chiavi. Utilizzare lo strumento di gestione chiavi per creare un file truststore denominato `scserver.cert`. Verificare che il file contenga il certificato.

Per creare un file truststore:

- 1 Creare un nuovo file con un editor di testo.
- 2 Digitare nel file quanto segue:


```
Truststore=sslclient.ini
Truststorepass=password
```
- 3 Salvare il file come `sslclient.ini` nella seguente directory:


```
TOMCAT_HOME/webapps/<NOME_SERVER_HUB>/WEB-INF
```
- 4 Chiudere il file.

Verifica SSL per server e client

Per verificare SSL:

- 1 Dal menu **Start** di Windows, scegliere **Programmi** > **Peregrine ServiceCenter** > **Client Java**.
- 2 All'avvio del client Java, verificare la presenza dell'icona del lucchetto e chiave nell'angolo in basso a destra della finestra del client Java stesso. Se l'icona è visualizzata, SSL è abilitato per il client specifico.
- 3 Accedere come falcon (o qualsiasi altro operatore valido).
- 4 Sul server, aprire il file registro del servlet del server Java.
L'ubicazione del file registro varia a seconda del programma di esecuzione del servlet del server Java e della sua configurazione. Per ulteriori informazioni, vedere *Risoluzione dei problemi* a pagina 117.
- 5 Nel file registro, cercare:
secure socket
- 6 Leggere il testo della registrazione relativa al secure socket. Il testo deve indicare che l'accesso, in un passaggio precedente, è riuscito e che SSL è abilitato.

Pacchetti di crittografia predefiniti

Un pacchetto di crittografia consiste in una combinazione di parametri crittografici che definiscono gli algoritmi di protezione e le dimensioni delle chiavi utilizzate per autenticazione, concordanza chiavi, cifratura e protezione dell'integrità. Le librerie SSL forniscono il supporto per la negoziazione di pacchetti di crittografia, come parte dell'handshaking SSL che avvia o verifica la comunicazione protetta. Il supporto SSL di ServiceCenter comprende i seguenti pacchetti di crittografia (elencati in ordine di preferenza).

```
SSL_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_DES_CBC_SHA SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
```


5 ServiceInfo Universal

CAPITOLO

ServiceInfo Universal (SIU) è un servlet che visualizza e aggiorna un modulo di ServiceCenter. Lo stesso risultato può essere ottenuto senza utilizzare servlet mediante la funzione Pubblicazione e abbonamenti. Per ulteriori informazioni, vedere Pubblicazione e abbonamenti nella *System Tailoring Guide, Volume 1* (in inglese).

Leggere questo capitolo per ottenere informazioni su

- *Requisiti dei servlet SIU* a pagina 112
- *Utilizzo di servlet Java* a pagina 112
- *Parametri SIU* a pagina 113
- *Parametri del file HTML* a pagina 114

Requisiti dei servlet SIU

Per l'installazione e l'esecuzione del servlet SIU sono richiesti quattro componenti:

- Un server Web oppure un motore servlet autonomo
- Motore servlet
- Client Java
- Server di ServiceCenter

Durante l'installazione standard del client Java viene installato anche il file del servlet SIU. Consiste di un unico file JAR Java: `serverhb.jar`. Il percorso predefinito di questo file è `\\Peregrine\ServiceCenter\java\serverhb.jar`. La configurazione della connessione SIU prevede tre passaggi:

- Configurare il server Web per le zone servlet oppure il motore servlet utilizzato deve essere in grado di avviare servlet senza ricorrere a un server HTTP.
- Configurare il motore servlet in modo che includa il file `serverhb.jar`.
- Configurare il client Java di ServiceCenter per il collegamento al SIU.

Il pacchetto servlet SIU passato al motore servlet è `com.peregrine.hub.sc.SIUServlet`. Per il nome di questo pacchetto viene fatta distinzione tra maiuscole e minuscole.

Utilizzo di servlet Java

Il SIU è un servlet Java. I servlet Java sono programmi Java sul lato server in grado di generare contenuti dinamici. Supportano un modello richiesta/risposta comunemente utilizzato dai server. Sono più efficienti dei programmi CGI che avviano un nuovo processo ogni volta che vengono richiamati. Dal lato client, l'accesso a un servlet Java è analogo a quello di uno script CGI standard. I servlet Java sono indipendenti dalla piattaforma, nonché più veloci e sicuri degli script CGI.

Parametri SIU

Di seguito sono illustrati i parametri di connessione del SIU, che si applicano a tutti i motori servlet. Questi parametri si applicano alla servlet del SIU e sono impostati durante la configurazione del motore servlet.

La seguente tabella elenca i parametri di connessione SIU relativi ai servlet SIU. Questi parametri possono essere impostati durante la configurazione del motore servlet. Perché il SIU funzioni occorre specificare tutti i parametri elencati, rispettando la distinzione tra maiuscole e minuscole..

Parametro	Descrizione
Heartbeat	Specifica la frequenza in secondi con la quale il SIU deve verificare la presenza di aggiornamenti sul server di ServiceCenter. Specificare un valore numerico in secondi. Ad esempio: Heartbeat=25
Timeout	Specifica il tempo di attesa di una risposta dal server di ServiceCenter prima di chiudere la sessione. Specificare un valore numerico in secondi. Ad esempio: Timeout=25
Verbose	Specifica se il servlet deve registrare o meno le informazioni sugli errori. I valori possibili sono True e False. Ad esempio: Verbose=25

Per un esempio di configurazione di un motore servlet, vedere *Esempio con Tomcat* a pagina 85 oppure *Esempio con JRun 3.1* a pagina 88.

Parametri del file HTML

Il client Java deve riconoscere il SIU per potervi stabilire una connessione. La seguente tabella elenca i parametri di connessione che possono essere inseriti nel file HTML utilizzato per avviare il client Java (`scjava.htm`, `scjavamac.htm` oppure `scjavaplugin.htm`). Se il client Java è di tipo autonomo, i parametri possono essere passati anche sulla riga di comando.

Nota: A differenza del server hub, il servlet SIU necessita dei parametri `Host` e `Service` definiti nel file HTML utilizzato per la connessione al SIU.

Parametro	Descrizione
HubURL	Specifica l'URL del SIU. Ad esempio: <code>http://www.host.com:8001/servlet/siu</code>
SCAdapter	Specifica il tipo di connessione da instaurare con il server. È necessario specificare <code>SCEXpress</code> .
SIU	Specificare il nome del modulo da visualizzare al momento della connessione. Ad esempio: <code>pm.status</code>
Host	Specifica l'indirizzo IP host del server di ServiceCenter. Questo indirizzo deve essere accessibile dai computer degli utenti.
Service	Specifica la porta del server di ServiceCenter.

Esempio di file HTML

Specificare i seguenti parametri nel file HTML usato per avviare il client Java. Questi parametri consentono l'esecuzione del client Java come applet e la connessione al server di ServiceCenter attraverso il SIU.

```
<param name="HubURL" value="http://www.host.com:8001/servlet/siu">
<param name="SCAdapter" value="SCEXpress">
<param name="siu" value="modulo_da_visualizzare">
<param name="Host" value="nome_host_ServiceCenter">
<param name="Service" value="modulo_da_visualizzare">
```

dove `www.host.com:8001/servlet/siu` va sostituito con l'URL del servlet SIU installato nel sistema in uso.

**Esempio
relativo alla
applicazione
autonoma**

Quando il client Java viene eseguito in modalità autonoma, i parametri possono essere passati al client sotto forma di argomenti della riga di comando.

```
-HubURL=http://www.host.com:8001/servlet/siu -siu=display_form  
-HubAdapter=SCExpress -Host:host.company.com -Service:12670
```

I parametri della modalità autonoma possono essere specificati anche nel file scj.ini.

6 Risoluzione dei problemi

CAPITOLO

È importante risolvere i problemi quando si verificano. Alcuni problemi sono più frequenti di altri. Leggere questo capitolo per ottenere informazioni sulla risoluzione dei problemi relativi a:

- *Sistemi Windows* a pagina 118
- *Sistemi Unix* a pagina 120
- *Sistemi Macintosh* a pagina 121
- *Server Hub* a pagina 122
- *Tutti i sistemi* a pagina 123

Sistemi Windows

In questa sezione vengono trattati i problemi che possono verificarsi con il sistema operativo Windows.

Esecuzione del client Java in un browser

Se si registrano problemi nell'esecuzione del client Java in un browser in ambiente Windows, Peregrine Systems consiglia di scaricare dal sito Web di Microsoft la JVM più recente.

Dipendenza client/server

Il client Java può essere aggiornato indipendentemente da ServiceCenter. Se si registrano problemi con il client Java, è possibile eseguire l'aggiornamento all'ultima versione senza aggiornare il server o le applicazioni di ServiceCenter.

- Il client Java 3.0 SP3 di ServiceCenter è compatibile con tutti i server versione 3.0.
- Il client Java versione 4.0 di ServiceCenter è compatibile con tutti i server 3.0 e 4.0.
- Il client Java versione 5.0 di ServiceCenter è compatibile con tutti i server versione 3.0, 4.0, 5.0 e 5.1.

Impossibile avviare sessioni multiple del client Java

Microsoft Internet Explorer 5.0 non consente di avviare più di un client browser alla volta. Per avviare più client browser è necessario utilizzare Internet Explorer 5.5 o superiore.

Configurazione della applet

Quale ausilio nella risoluzione dei problemi relativi all'esecuzione, il client genera dei messaggi di registrazione dell'attività, visualizzabili nella console Java del browser.

- Per visualizzare il registro dei messaggi in Internet Explorer, deve essere attivata la registrazione dell'attività Java.
 - Selezionare **Opzioni Internet nel menu Strumenti**.
 - Fare clic sulla scheda **Avanzate**.
 - Scorrere verso il basso fino a **Microsoft VM**.
 - Selezionare **Registrazione attività Java attivata e Console Java attivata**.
 - Fare clic su **OK**. Riavviare il browser.
 - Selezionare **Console Java nel menu Strumenti**.
- Per visualizzare i messaggi di registrazione in Netscape Navigator 4.5, scegliere **Console Java nel menu Communicator->Strumenti**.

Per informazioni sulla compatibilità del browser, vedere *Sito Web CenterPoint di Peregrine* a pagina 9.

Configurazione dell'applicazione autonoma

Quale ausilio nella risoluzione dei problemi relativi alle applicazioni autonome, il client memorizza in un file denominato `scj.log` dei messaggi di registrazione dell'attività. Selezionare **Informazioni su ServiceCenter** dal menu **Guida** nella barra dei menu del client Java per visualizzare il percorso del file registro nel sistema in uso.

Se non è possibile avviare il client Java, accertarsi che sia stata configurata una connessione client con il server di ServiceCenter. È possibile utilizzare un client Windows standard per convalidare la connessione al server.

Parametro MaxChars

Il parametro `MaxChars`, che può essere impostato nella finestra delle proprietà di Progettazione moduli, è imposto dal client Java. L'impostazione predefinita è 0. Se si modifica questa impostazione, possono risultare stringhe di testo troncate negli oggetti che utilizzano questo parametro.

La soluzione consiste nel modificare l'impostazione nella finestra Proprietà specificando un valore più alto oppure nel riportarla al valore predefinito 0.

Sistemi Unix

In questa sezione vengono trattati i problemi che possono verificarsi con il sistema operativo Unix.

Copia e incolla (tutti)

I computer Unix/CDE dispongono generalmente di due buffer di selezione, PRIMARY e CLIPBOARD. Il testo evidenziato è normalmente inserito nel buffer PRIMARY, dal quale può essere incollato in altre applicazioni utilizzando il pulsante centrale del mouse. Purtroppo Java attualmente supporta unicamente il buffer CLIPBOARD per le operazioni di taglia e incolla. Per aggirare il problema può essere utilizzata un'applicazione intermedia (come dtterm o xclipboard) in grado di accedere a entrambi i buffer.

Per copiare testo da xterm nel client Java:

- 1 Selezionare il testo nella finestra di xterm.
- 2 Utilizzare il pulsante centrale del mouse per incollare il testo in una finestra dtterm.
- 3 Selezionare il testo nella finestra di dtterm. Selezionare **Copy** dal menu **Edit**.
- 4 Posizionare il cursore nel componente di testo desiderato del client Java. Selezionare **Incolla** dal menu **Modifica**.

Per copiare testo dal client Java a xterm (procedura inversa):

- 1 Selezionare il testo nel client Java. Selezionare **Copia** dal menu **Modifica**.
- 2 Posizionare il cursore nella finestra di dtterm. Selezionare **Paste** dal menu **Edit**.

Per copiare testo da dtterm a xterm:

- 1 Selezionare il testo nella finestra di dtterm.
- 2 Utilizzare il pulsante centrale del mouse per incollare il testo nella finestra di xterm.

HP-UX 11 e Netscape

Gli utenti che utilizzano Netscape su sistemi HP-UX 11 devono avviare il browser con l'opzione -visual Default. Ad esempio:

```
netscape -visual Default
```

Linux

La funzione di copia e incolla non è eseguibile se l'applicazione non condivide la stessa area di appunti del client Java.

Client (tutti)

Se si desidera eseguire il client Java basato su browser su un sistema Unix, è necessario poter avviare la console Java. Se la console non viene visualizzata, la Java VM non è configurata correttamente e non è accessibile.

Consultare la documentazione del browser per accertarsi che le classi Java siano accessibili. In alcuni casi ciò significa reimpostare l'ambiente CLASSPATH. Per Netscape Navigator, impostare la variabile di ambiente MOZILLA_HOME indicando il percorso di base (directory) del browser.

Se si intende eseguire un client Java autonomo, alcuni sistemi funzionano meglio con determinati JRE che con altri. Se il client Java provoca un'eccezione o genera errori, installare una versione più recente del JRE oppure tornare a una precedente versione stabile del JRE.

SUN OS

Viene generata un'eccezione quando si tenta di avviare il client Java come utente e non si è avviata la sessione corrente di X Window.

L'utente del client Java deve essere il proprietario della sessione corrente di X Window. È possibile disabilitare la protezione xhost utilizzando il comando `xhost +;` in tal modo gli utenti di qualsiasi sistema possono accedere all'ambiente X e gestirlo, ma la protezione risulterà ridotta.

Sistemi Macintosh

cannot connect null:null

Se non si riesce ad avviare il client Java come applicazione autonoma o si ricevono messaggi di errore che indicano l'impossibilità di connettersi (cannot connect null:null), memorizzare il file `scj.ini` nella directory delle preferenze e riavviare il client.

MRJ 2.2.4

Il problema consiste nel fatto che con il tasto TAB è possibile spostarsi solo tra il campo di indirizzo del browser e il primo campo del modulo visualizzato nel client Java. È un problema noto di Java con Internet Explorer sui sistemi Macintosh. Il browser elabora tutti gli eventi del tasto TAB senza sottoporli alla applet Java eseguita dal browser. Ciò vale per il client Java di ServiceCenter e per tutte le altre applet Java.

Server Hub

Il server hub genera dei messaggi diagnostici di base utili per investigare i problemi che si verificano. Esaminare il file registro quale ausilio nella risoluzione dei problemi di configurazione. Il file registro può essere inviato tramite posta elettronica al servizio assistenza di Peregrine Systems, Inc. accompagnato dalla descrizione dettagliata degli eventi che hanno generato il problema.

Il percorso del file registro varia a seconda del motore servlet utilizzato e della relativa configurazione. Ad esempio, se si utilizza il server JRun con le impostazioni predefinite del file registro, il file si trova in:

```
JRun_Home/jsm-default/logs/stdout.log
```

dove JRun_Home è la directory principale dell'installazione di JRun.

Parametro UseGetMethod

Questo parametro serve per la connessione al server hub ed è necessario solo per determinate combinazioni di browser Web e motori servlet. Non deve essere specificato, a meno che non si abbiano problemi nella connessione al servlet da un browser, e si sia certi che il servlet è configurato correttamente. Il parametro UseGetMethod può assumere il valore true o false. Il valore predefinito è false.

Se per il parametro UseGetMethod è impostato su true, il client tenterà di connettersi al servlet con il metodo HTTP GET, altrimenti utilizzerà il metodo HTTP POST. La sintassi per questo parametro è la seguente:

```
<param name="UseGetMethod" value="true">
```

File HTML

Non inserire barre rovesciate (\) nell'URL HTTP. Utilizzare solo la barra /.
Ad esempio l'espressione seguente è valida:

```
<param name="HubURL" value="http://localhost:8001/servlet/hub">
```

L'espressione seguente non è valida e genererà un messaggio di errore alla connessione:

```
<param name="HubURL" value="http://localhost:8001\servlet\hub">
```

Tutti i sistemi

Trascinamento

Il client Java non supporta operazioni di trascinamento. Gli allegati devono essere inseriti mediante i comandi di menu.

7 Specifiche di accessibilità

CAPITOLO

In seguito all'aggiunta della Section 508 al Rehabilitation Act del 1973, gli enti governativi federali statunitensi sono tenuti a impiegare prodotti e servizi informatici ed elettronici accessibili anche a utenti disabili. Grazie alla conformità alla Section 508 che caratterizza il client Java, ServiceCenter è in grado di soddisfare i requisiti fissati da tale normativa.

La libreria Java Swing è uno strumento di sviluppo per interfacce grafiche utente che fa parte della piattaforma Java 2 Standard Edition (J2SE). Questa libreria offre supporto diretto per l'implementazione di funzioni di accessibilità nel prodotto software finale. Queste funzioni di accessibilità contenute nel software consentono ai produttori di tecnologie per disabili di personalizzare l'interfaccia del software stesso affinché supporti formati di presentazione diversi da quelli visivi, mediante dispositivi audio o di altro tipo. Ciò assicura agli utenti del client Java la possibilità di configurare l'interfaccia in base alle proprie esigenze nell'applicazione di ausili tecnologici di terzi.

Leggere questo capitolo per ottenere informazioni su

- *Caratteristiche di conformità alla Section 508* a pagina 126
- *Ausili tecnologici* a pagina 127
- *Impostazione delle opzioni di visualizzazione* a pagina 129
- *Impostazione delle opzioni di modifica* a pagina 129
- *Aspetti relativi alla conformità alla Section 508* a pagina 130

Caratteristiche di conformità alla Section 508

La conformità alla Section 508 della normativa statunitense è assicurata dalle seguenti caratteristiche:

- Quasi tutti i campi di testo e gli oggetti nel client Java dispongono di un nome e di campi descrittivi.
- Progettazione moduli supporta l'assegnazione manuale di nomi e campi descrittivi ai restanti oggetti e campi di testo. Questa funzione è disponibile anche nei client Windows.

Per ulteriori informazioni su Progettazione moduli, consultare la *System Tailoring Guide, Volume 1* (in inglese).

- Per i componenti di sola lettura e gli altri componenti obbligatori, al campo Nome accessibile vengono aggiunte stringhe di caratteri leggibili ad alta voce con gli ausili tecnologici di terzi.
- È possibile salvare nel file di inizializzazione del client Java le proprie preferenze relative all'accesso da tastiera, al lampeggiamento di segnalazione, al tipo di carattere, all'aspetto grafico e ai colori.
- È disponibile un'opzione di riga di comando che, prima di accedere al client Java, abilita le preferenze definite in precedenza per un determinato utente.

Funzioni da tastiera

Queste funzioni migliorano l'accesso da tastiera al client Java. È possibile:

- Definire un accesso da tastiera opzionale, tramite il tasto TAB, a campi di sola lettura, barra di stato, finestre di messaggio e personalizzare lo spostamento tramite tasto TAB nei campi delle tabelle.
- Selezionare un'opzione di menu che abiliti o disabiliti l'accesso da tastiera a tutti i campi.
- Selezionare un'opzione di menu che consenta di sostituire lo spostamento predefinito da sinistra a destra mediante il tasto TAB con lo spostamento dall'alto verso il basso.

Preferenze di visualizzazione

Queste funzioni migliorano la visualizzazione nel client Java. È possibile:

- Selezionare un'opzione di menu per abilitare o disabilitare il lampeggiamento dei componenti, ad esempio l'icona Attività.
- Accedere a una finestra di dialogo di selezione del tipo di carattere per modificare la famiglia di font, le dimensioni e l'evidenziazione.
- Selezionare un'opzione di menu per modificare l'aspetto grafico del client Java.
- Accedere a una finestra di dialogo di definizione e selezione dello schema di colori.

Ausili tecnologici

Per rispondere ai requisiti della Sezione 508, nel client Java sono state implementate funzionalità che consentono di utilizzare tecnologie e strumenti per disabili sviluppati da terze parti. Si tratta di componenti di accessibilità standard che ampliano le funzionalità a disposizione degli utenti disabili.

Strumenti per utenti offerti da terze parti

Diversi produttori di ausili tecnologici sono specializzati nella realizzazione di software e hardware specificamente studiati per le esigenze degli utenti. Alcuni dispositivi hardware, quali workstation Braille e tastiere con tasti più grandi, e prodotti software, quali utilità di riconoscimento vocale e di ingrandimento dello schermo, sono in grado di migliorare l'interazione tra utente e software. Alcuni strumenti software sono in grado di leggere il contenuto della finestra corrente, ad esempio JAWS di Freedom Scientific. Altri strumenti di riconoscimento vocale possono essere utilizzati dall'utente per ottenere un testo tramite conversione della dettatura. Due esempi sono rappresentati da Via Voice di IBM e Dragon Naturally Speaking di ScanSoft.

Strumenti di sviluppo

Gli sviluppatori Java utilizzano questi strumenti per consentire un'interazione ottimale tra il software, il sistema operativo e i prodotti di accessibilità di terze parti.

Java Accessibility API

La Java Accessibility API è un'interfaccia di sviluppo che fornisce informazioni e contenuti per tecnologie ausiliarie per disabili quali Jaws for Windows.

Pacchetto Java Accessibility Utility

Il pacchetto Java Accessibility Utilities consente ai produttori di tecnologie ausiliarie per disabili di individuare ed accedere a tutti componenti, quali campi, pulsanti, caselle di controllo e pulsanti di opzione, di un'applicazione Java eseguita in ambiente JVM.

Java Accessibility Bridge

Java Accessibility Bridge collega il software eseguito in ambiente JVM al software in esecuzione su una piattaforma nativa ed è specifico per ciascuna piattaforma. Si tratta di un programma bridge che consente al client Java di sfruttare le funzionalità del sistema operativo nativo specifiche per l'accessibilità e la comunicazione con ausili di terze parti.

Nota: Questo programma bridge è attualmente disponibile solo per il sistema operativo Windows.

Impostazione delle opzioni di modifica

È possibile impostare le preferenze per la disattivazione delle animazioni e l'accesso a tutti i componenti: campi (anche a sola lettura), pulsanti, caselle di controllo e pulsanti di opzione.

Per impostare le preferenze relative all'accessibilità e alle animazioni:

- 1 Scegliere **Preferenze > Accessibilità > Accedi a tutti i campi** dal menu **Modifica** per abilitare gli ausili tecnologici per la lettura di testo, quali JAWS, che consentono di leggere gli elementi visualizzati nelle finestre di ServiceCenter (campi, pulsanti, caselle di controllo e pulsanti di opzione).
- 2 Dal menu **Modifica**, scegliere **Preferenze > Accessibilità > Ignora ordine di tabulazione server**. Se viene selezionata l'opzione Ignora ordine di tabulazione server, lo spostamento tramite tasto TAB avviene dall'alto verso il basso.
- 3 Dal menu **Modifica**, scegliere **Preferenze > Accessibilità > Disattiva animazione** per disabilitare il lampeggiamento della spia di stato e dei messaggi nella barra di stato.

Impostazione delle opzioni di visualizzazione

L'architettura "Pluggable look and feel" consente al client Java di emulare l'interfaccia dei sistemi operativi Windows e Unix Motif e di offrire schemi di colori opzionali. Questa funzione consente inoltre ai produttori di tecnologie ausiliarie per disabili di implementare caratteristiche di personalizzazione dell'interfaccia per presentazioni audio o dispositivi hardware speciali.

Per impostare l'aspetto e il funzionamento dell'interfaccia:

- Dal menu **Modifica**, scegliere **Preferenze > Aspetto e funzionamento**. È possibile scegliere **Metal**, **CDE/Motif** oppure **Windows** per modificare l'aspetto complessivo dell'interfaccia del client Java.

Per impostare uno schema di colori:

Dal menu **Modifica**, scegliere **Preferenze > Schema di colori**. La Figura 7-1 mostra due menu a discesa che consentono di scegliere un diverso schema di colori e di visualizzare come tale schema si applichi ai diversi elementi dell'interfaccia.

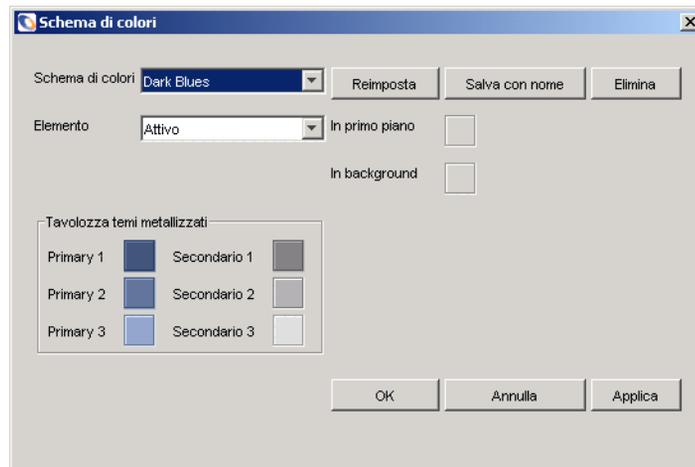


Figura 7-1: Schemi di colori

Aspetti relativi alla conformità alla Section 508

Alcuni aspetti della versione 5.1 del client Java potrebbero limitare le possibilità di scelta a disposizione dell'utente. Alcune funzionalità non sono ancora disponibili e vi sono dei problemi noti riguardanti strumenti di accessibilità di terzi.

Problemi noti

Alcuni problemi limitano la possibilità di utilizzo delle funzioni di accessibilità da parte degli utenti del client Java.

- La versione 4.5 del software JAWS non è in grado di leggere i dati contenuti nelle tabelle.
- JAWS legge solo il nodo più alto della struttura di Explorer e non è in grado di riconoscere la modifica del nodo selezionato.
- Alcuni moduli contengono campi o nomi di oggetti cui non è possibile assegnare un nome accessibile. È possibile assegnare il nome manualmente utilizzando Progettazione moduli.
- L'utilità Progettazione moduli non è conforme alla Section 508. Sarà l'amministratore del sistema a dover apportare le necessarie modifiche ai moduli per soddisfare le esigenze dell'utente.

Funzioni non disponibili

Alcune funzionalità non sono disponibili nella presente versione di accessibilità del client Java.

- Accessibilità al testo scorrevole.
- Accessibilità ai nomi e alle descrizioni delle illustrazioni.
- Accessibilità alternativa ai grafici.

Indice

A

Abstract Window Toolkit (AWT) 14
accessibilità
 abilitazione delle tecnologie ausiliarie 129
 animazione 129
 API 128
 aspetti relativi alla conformità 130
 aspetto grafico e funzionamento 129
 Java Accessibility API 128
 Java Accessibility Bridge 128
 Java Accessibility Utilities 128
 JAWS 131
 libreria Java Swing 125
 opzioni di modifica 129
 opzioni di visualizzazione 129
 ordine di tabulazione 129
 preferenze di visualizzazione 127
 problemi noti 131
 Progettazione moduli 131
 Rehabilitation Act del 1973 125
 requisiti 125
 schema di colori 130
 tastiera 126
aggiornamento, client Java 31
Apache Tomcat 85, 102
AppArgs, parametro 42
Apple SimpleText, editor 51
applet, configurazione, risoluzione dei
 problemi 119
applet, risoluzione dei problemi di
 configurazione 119

args, parametro 43
assistenza clienti 9
assistenza tecnica 9
AWT 14

B

barra degli strumenti, Preferiti 12
barra di stato 12
basato su browser, client Java
 esecuzione 37, 53
bitmap, file 37
bitmaps.zip, file 37
browser 16, 36

C

cache, directory 37
cartella di destinazione 24
CenterPoint, sito Web 8, 9
chiave
 predefinita 103, 106
 privata 103, 107, 108
chiave predefinita 106
classi, Java 36
classpath 75, 87
client Java
 allegare file 13
 ambiente di runtime
 vedere JRE
 animazione 129
 applicazione browser 32, 33
 aspetto grafico e funzionamento 129

- autonomo 17, 20, 21, 31, 75, 119
 - basato su browser 16, 106, 108, 121
 - caratteristiche 12
 - configurazione 75
 - console 119
 - dimensioni heap 45
 - elenchi di record, aggiornamento 76
 - esecuzione 37, 53
 - file applet 36
 - file immagine 36
 - installazione 17
 - Macintosh 49, 51, 53
 - nome del servizio 76
 - opzioni di visualizzazione 129
 - ordine di tabulazione 129
 - OS/2 53
 - parametri 62
 - personalizzazione dell'interfaccia 13
 - porta di connessione 90
 - pulsanti dei grafici 14
 - schema di colori 130
 - servlet 112
 - stampa 13
 - test della connessione di rete 55
 - virtual machine 18
 - client Java autonomo 17, 20, 21, 31, 53, 75
 - client Java basato su browser
 - chiave predefinita 106
 - chiave privata 108
 - installazione 15, 32
 - locale 32
 - relazione con il server 16
 - remoto 33
 - Unix 121
 - ClientPort, parametro 92
 - clientprinting, parametro 66, 70
 - codebase, variabile 41
 - codeset, parametro 62
 - collegamenti ipertestuali 12
 - comandi, setup 21
 - conformità alla Section 508
 - vedere accessibilità
 - ConnectionType, parametro 98
 - connectport, parametro 82
 - connessione di rete, test 55
 - connessione diretta, server hub 102, 103
 - console
 - Java 55, 56
 - ServiceCenter 31
- D**
- default, parametro 81
 - dictionarydir, parametro 62
 - dinamico, indirizzo IP 84
 - downloaddictionarydir, parametro 62
 - dtterm, applicazione 120
- E**
- Education Services 10
 - elenchi di record, aggiornamento automatico 76
 - elenco di record, frequenza di aggiornamento
 - automatico 76
 - envdump, parametro 62
 - etc/services, file 41
 - explorer, parametro 66
 - explorerdefault, parametro 67
 - explorerhome, parametro 67, 70, 74
- F**
- file
 - bitmap 37
 - bitmaps.zip 37
 - etc/services 41
 - HTML 35
 - Java client applet 36
 - Java client images 36
 - jbird.jar 75
 - jcrt.jar 105
 - jnet.jar 105
 - jsse.jar 105
 - parametri della riga di comando scjava 42
 - sc.ini 70
 - scapplet.htm 17, 38
 - scj.ini 50, 63, 121
 - scj.log 119
 - scjava 45
 - scjava.htm 16, 38, 82, 90, 114
 - scjava13plugin.htm 38, 55
 - scjava13launch.htm 16, 38, 53, 54
 - scjavamac.htm 16, 53, 82, 90, 114

- scjavaplugin.htm 82, 90, 114
- scjpref.ini 62
- scjversions.properties 37
- scmac.htm 38
- scoicon.exe 37
- scssl.ini 102
- serverhb.jar 87, 88
- setup.exe 21
- sslserver 104
- truststore 108
- web.xml 86, 103
- firewall 90, 91, 96, 101
- frequenza di aggiornamento automatico, elenco di record 76

H

- heap client Java, dimensioni 45
- Heartbeat, parametro 113
- host, parametro 43, 76, 114
- HP-UX 11 120
- HTML
 - descrizioni dei file 35
 - esempio di file 91, 93, 95, 98, 101
 - esempio di file SIU 114
 - pagina 60
 - parametri 82, 114
- HTTP
 - connessione 84
 - prefisso 34
 - server 84
- hub, directory 86
- HubAdapter, parametro 83, 91, 92, 97
- HubURL, parametro 82, 97, 114

I

- ID servizio, TCP/IP 26, 41, 76
- imagepath, parametro 37, 76
- images, parametro 43
- IME 12
- indirizzo del servizio, TCP/IP 15, 18
- indirizzo IP
 - dinamico 84
 - statico 84

- installazione
 - client Java autonomo
 - Unix 40, 41
 - Windows 20
 - client Java basato su browser (Unix) 41
 - client Java OS/2 53
 - Macintosh
 - client Java per OS 9 49
 - client Java per OSX 51
 - MRJ 46
 - multipla 22
 - personalizzata 23
- installazione OS/2 53
- installazione personalizzata 23
- Internet Explorer, Microsoft 16
- ISO, codici di lingua 77

J

- Java
 - classpath 87, 88
 - console 55, 56
 - Foundation Classes (JFC) 14, 36
 - libreria Swing 14, 37, 125
 - Runtime Environment
 - vedere JRE
 - Secure Socket Extensions (JSSE) 105
 - Virtual Machine (JVM) 18
- java, parametro 43
- JAWS 131
- jbird.jar, file 75
- jcrt.jar, file 105
- JFC 14
- jnet.jar, file 105
- JRE
 - esecuzione di un client autonomo 18
 - strumento di gestione chiavi 104
 - Sun Microsystems 18
 - versione 1.1.8 54
 - versione 1.2.2_008 20
 - versione 1.3x 20, 38, 54
 - versioni precedenti 18
- JRun 88
- jsse.jar, file 105
- JView JRE, Microsoft 18
- JVM 18, 54

L

language, parametro 62
 lib, directory 86
 lingue, codici 77

M

Macintosh
 esecuzione di un client Java 53
 installazione del client Java 49, 51
 MRJ 46
 risoluzione dei problemi 121
 scjava.launch.htm 38
 MaxChars, parametro 119
 MDI 12
 menuforms, parametro 67
 Microsoft
 Input Method Editor (IME) 12
 Internet Explorer 16
 JView JRE 18
 MRJ 46, 122
 multiple
 installazioni 22
 sessioni 12
 Multiple Document Interface
 vedere MDI

N

NAT 84
 NAT (Network Address Translation)
 vedere NAT
 Netscape Navigator 16, 36
 nohelpontfield, parametro 67
 nome host, TCP/IP 15, 18
 Note attive, finestra 55

O

Object Linking and Embedding (OLE) 13
 Opzione Stampa sul client 70

P

pacchetti di crittografia 109
 pacchetti di crittografia predefiniti 109
 parametri
 AppArgs 42
 args 43
 ClientPort 92
 clientprinting 66, 70
 codeset 62
 ConnectionType 98
 connectport 82
 default 81
 di controllo client/server Java 66
 dictionarydir 62
 downloaddictionarydir 62
 envdump 62
 explorer 66
 explorerdefault 67
 explorerhome 67, 70, 74
 Heartbeat 113
 host 43, 76, 114
 HTML 82, 114
 HubAdapter 83, 91, 92, 97
 HubURL 82, 97, 114
 imagepath 37, 76
 images 43
 java 43
 language 62
 MaxChars 119
 menuforms 67
 nohelpontfield 67
 SCAdapter 114
 scjpath 62
 sctimeramount 67
 Server 83, 91, 92, 98
 server hub 81
 servername 96
 servers 81
 service 43, 76, 114
 SIU 113, 114
 threads 82
 Timeout 113
 Verbose 113
 viewactivenotes 67
 parametri della riga di comando 42, 75

- Peregrine Systems
 - assistenza clienti 9
 - CenterPoint, sito Web 8
 - informazioni di contatto in tutto il mondo 9
 - Sede centrale 9
- porta di connessione 98
 - client Java 90
 - diretta 98
- porta, numero 22, 26
- predefinita, chiave 103
- predefiniti, pacchetti di crittografia 109
- Preferiti, barra degli strumenti 12
- privata, chiave 103, 107, 108
- Progettazione moduli
 - proprietà 119
 - utilità 76, 131
- Pubblicazione e abbonamenti, funzione 111
- pulsanti dei grafici, client Java 14

R

- Ripristina moduli all'avvio, opzione 12
- risoluzione dei problemi
 - client Java autonomo 119
 - configurazione delle applet 119
 - copia e incolla 120
 - HP-UX 11 120
 - Macintosh 121
 - MRJ 122
 - server hub 122
 - sistemi Windows 118
 - trascinamento 123
 - Unix 120

S

- salvataggio all'uscita 12
- sc.ini, file 70
- SCAdapter, parametro 114
- scapplet.htm, file 17, 38
- SCEXpress 91, 97, 114
- SCEXpress, connessione 83
- SCEXpressSL, connessione 83
- scj.ini, file 50, 63, 121
- scj.log, file 119
- scjava, file 42, 45
- scjava.htm, file 16, 38, 82, 90, 114

- scjava13plugin.htm, file 38, 55
- scjavaLaunch.htm, file 16, 38, 53, 54, 60
- scjavamac.htm, file 16, 53, 82, 90, 114
- scjavaplugin.htm, file 82, 90, 114
- scjpath, parametro 62
- scjpref.ini, file 62
- scjversions.properties, file 37
- scmac.htm, file 38
- scoicon.exe, file 37
- scssl.ini, file 102
- sctimeramount, parametro 67
- Section 508, conformità
 - vedere accessibilità
- server hub
 - client Java 91
 - connessione callback 83, 84, 92
 - connessione diretta 83, 102, 103
 - connessione HTTP 84
 - connessione TCP/IP 84
 - definizione di servlet 80
 - esempio relativo al client autonomo 93, 96, 98, 101
 - file registro 122
 - pacchetto servlet 80
 - parametri di connessione 81
 - risoluzione dei problemi 122
 - server proxy 84
 - supporto di servlet 81
- server proxy 84
- Server, parametro 83, 91, 92, 98
- serverhb.jar, file 87, 88
- servername, parametro 96
- servers, parametro 81
- service, parametro 43, 76, 114
- ServiceCenter
 - console 31
 - Explorer 12, 13
- ServiceInfo Universal (SIU)
 - vedere SIU
- servizi di formazione 10
- servlet
 - motore 97
 - zone 97
- setup, comando 21
- setup.exe, file 21

SIU

- definizione 111
- esempio di configurazione autonoma 115
- esempio di file HTML 114
- pacchetto servlet 112
- parametri di connessione 113
- parametro 114
- servlet client Java 112

SSL

- algoritmo crittografico 102
- certificato 102
- descrizione 102
- jnet.jar 105
- jsse.jar 105
- librerie 105
- pacchetti di crittografia 109
- scssl.ini, file 102
- strumento di gestione chiavi 104
- verifica 109
- web.xml, file 103
- sslserver.ini, file 104
- Stampa sul server 70
- statico, indirizzo IP 84
- strumento di gestione chiavi JRE 104
- Sun Microsystems 18
- supporto SSL (Secure Socket Layer)
 - vedere SSL
- Swing, libreria Java 14, 37

T

- tastiera, conformità alla Section 508 126
- TCP/IP
 - comunicazione server 12
 - connessione 84
 - ID servizio 26
 - indirizzo del servizio 15, 18
 - nome host 15, 18
 - porta, numero 26
 - socket 12
- test della connessione di rete 55
- threads, parametro 82
- Timeout, parametro 113
- trascinamento 123
- truststore, file 108

U**Unix**

- buffer CLIPBOARD 120
- buffer PRIMARY 120
- client Java autonomo 41
- client Java basato su browser 41
- copia e incolla 120
- dtterm 120
- esecuzione di un client Java autonomo 42
- installazione di un client Java 40
- risoluzione dei problemi 120
- xclipboard 120
- URL 32, 41
- utilità, Progettazione moduli 76

V

- Verbose, parametro 113

versione

- JRE 1.1.8 54
- JRE 1.2.2_008 20
- JRE 1.3x 38, 54
- MRJ 2.2.4 122
- nuovo ServiceCenter 22
- viewactivenotes, parametro 67
- visualizzare/nascondere i comandi del browser 16

W

- web.xml, file 86, 103
- Web-inf, directory 86
- Windows

- client Java autonomo 20
- scjava.launch.htm, file 38
- sistema operativo XP 18
- XP, supporto Java 18

X

- xclipboard, applicazione 120
- XP, Windows 18

