ESPRITSYSTEMS

# Esprit Server
# System Konfiguration

**Konfiguration und Betrieb des Esprit Client/Server Systems**

**November 2014**
**Rainer Büsch**

# Inhaltsverzeichnis

# 1 Einführung

Dieses Dokument beschreibt die Inbetriebnahme eines Esprit Client/Server Systems und erläutert die Konfigurationsmöglichkeiten des Esprit-Servers.

## 1.1 EspritWorkbench-Projekt

Als Grundlage für die folgenden Erläuterungen dient das Eclipse-Projekt *EspritWorkbench*, das als Template für reale Kundenprojekte zur Verfügung steht. Im Folgenden wird davon ausgegangen, dass der Leser dieses Projekt in seiner Eclipse Entwicklungsumgebung lauffähig installiert hat.

## 1.2 Installation der EspritWorkbench

Downloaden Sie die Datei www.esprit-systems.de/downloads/esprit/EspritWorkbench.zip und extrahieren sie in Ihr Eclipse Workspace-Verzeichnis, z.B. *C:\Projects*.

Starten Sie *Eclipse* und erzeugen ein neues Java-Projekt namens *EspritWorkbench*. Eclipse entdeckt, dass ein gleichnamiges Projekt bereits existiert und öffnet es. Da das Projekt alles Notwendige beinhaltet, sollte es fehlerfrei kompilieren können. Eventuell muss der Java-Build-Path an das auf Ihrem System installierte JDK angepasst werden. Siehe Menü: *Project → Properties → Java build path*.

## 1.3 Vorkonfigurierte Launches

Im Verzeichnis *start* sind bereits Starteinträge für Server und Client vorkonfiguriert. Diese erscheinen nach dem Öffnen des Projekts auch im Eclipse-Run-Menü und können dort unmittelbar aufgerufen werden.

# 2 Starten des Esprit-Systems

## 2.1 Hochfahren des Servers

Starten Sie den Esprit-Server durch Klick auf *EwbServerLaunch* im Eclipse-Run-Menü. Der Server initialisiert und startet seine Netzwerkdienste. Beobachten Sie die Systemkonsole:

```
Starting up server <EWB-Server>... ... ...
INF: [EWB-Server] WebServer: Started listening on HTTP port: 9090
INF: [EWB-Server] IOServerConnectionService: Started listening on port: 2011
INF: [EWB-Server] EwbServerContext: Server <EWB-Server> is up and running
```

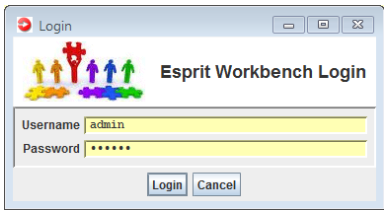
Ein Blick in den Launch *EwbServerLaunch* zeigt, dass er mit folgenden Optionen gestartet wurde:

- ➔ Option: **-workspace publish/server** (Pflicht)
  Weist dem Server sein Arbeitsdirectory zu. Achtung: Sie können einen absoluten oder relativen Pfad angeben, aber er muss als Directory existieren. Falls es leer ist, legt der Server dort Unterverzeichnisse an, insbesondere zur Speicherung seiner Konfigurationsdateien. Sollten diese bereits existieren, werden sie lediglich eingelesen.
- ➔ **VM**-Option: **-server** (optional)
  Startet die Java-VM im *server*-Mode. Sie läuft dann besser optimiert für Parallelanfragen.

➔ **VM**-Option: **-Djava.awt.headless=true** (Pflicht auf Rechnern ohne Grafikkarte)
Startet die Java-VM im *headless*-Mode. In diesem Mode wird kein Event-Dispatching Thread gestartet, der normalerweise das GUI betreibt. Er ist bei Hintergrundprozessen, wie dem Esprit-Server, nicht notwendig, da dieser keine Benutzeroberfläche besitzt.

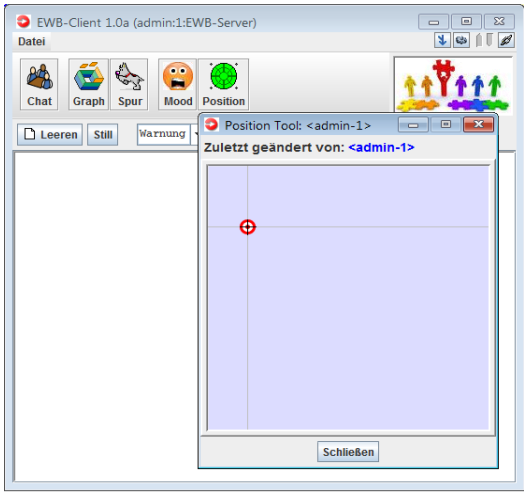
## 2.2 Starten des Clients

Starten Sie den Esprit-Client durch Klick auf *EwbClientLaunch* im Eclipse-Run-Menü. Der Client meldet sich mit seinem Login-Dialog.




Geben Sie als Benutzername *admin* und als Passwort *esprit* ein. Dies sind die Standard-Voreinstellungen, die Sie selbstverständlich später ändern sollten.

Nach erfolgreichem Login erscheint der Client-MainFrame. Nun können sie die verschiedenen Tools, z.B: das *Position Tool*, ausprobieren. Starten Sie bitte noch weitere Clients, in denen sie ebenfalls das *Position Tool* öffnen. Egal in welchem Client Sie das kleine Zielscheiben-Icon bewegen, Sie können die Änderung unmittelbar in allen anderen Clients sehen. Dies ist der „Esprit-Effekt“!




Ein Blick in den Launch *EwbClientLaunch* zeigt, dass er mit folgenden Optionen gestartet wurde:

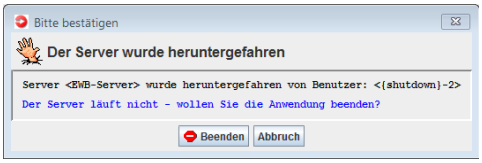
➔ Option: **-server localhost:2011** (Pflicht)
Definiert die Netzwerkadresse des Servers.

➔ Option: **-user admin -password esprit** (optional)
Definiert die Default-Einträge für Benutzer und Passwort im Login-Dialog.

➔ Option: **-autologin** (optional)
Bestätigt den Login-Dialog automatisch, falls Benutzer und Passwort angegeben wurden.

## 2.3 Herunterfahren des Servers (lokal)

Um den Serverprozess zu beenden, klicken Sie auf *EwbLocalServerShutdown* im Eclipse-Run-Menü. Der Server schließt alle seine Client-Verbindungen und terminiert dann selbst. Ein betroffener graut alle netzwerkbezogenen Aktionen aus. Zusätzlich erscheint ein Dialog, der Sie zum Beenden des Clients auffordert. Wenn Sie diesen Dialog abbrechen, bleibt der Client bestehen, allerdings ohne Serververbindung.




Nun können Sie den Server erneut hochfahren. Ein Klick auf den Connect-Knopf des Clients stellt die Serververbindung wieder her. Dies ist mit einer vollständigen Initialisierung des Clients verbunden, daher zeigt dieser stets die aktuellsten Daten.


Ein Blick in den Launch *EwbLocalServerShutdown* zeigt, dass dieser mit folgenden Optionen gestartet wurde:

➔ Option: **-workspace publish/server -shutdown** (Pflicht)
In Wirklichkeit entspricht dies dem erneuten Aufruf von *EwbServerLaunch* mit dem Unterschied, dass durch die **-shutdown** Option der laufende Server angesprochen und heruntergefahren wird.

## 2.4 Herunterfahren des Servers (remote)

Die oben gezeigte Art des Herunterfahrens funktioniert nur auf dem Host, auf dem der Server tatsächlich läuft. Um einen Server, der auf einem anderen Rechner (z.B. Hostname *orion*) läuft, herunterzufahren, rufen Sie ***EwbRemoteServerShutdown*** aus dem Eclipse-Run-Menü auf. Dieser Launch startet einen speziellen Client ohne Benutzeroberfläche, der sich mit dem Server verbindet, um ihn zu terminieren. Ein Blick in den Launch zeigt, dass er mit folgenden Optionen gestartet wird:

- Option: **-server orion:2011** (Pflicht)
  Definiert die Netzwerkadresse des Servers.
- Option: **-user admin -password esprit** (Pflicht)
  Definiert das Administrator-Login mit Benutzername und Passwort. Natürlich darf nur ein Administrator den Server herunterfahren.

# 3 Konfiguration des Esprit-Systems

## 3.1 Der Client Workspace

Ein Esprit-Client benötigt ein Arbeitsdirectory (seinen Workspace), wo er Daten speichern kann. Mit folgender Launch-Option kann ihm dieses zugewiesen werden:

- Option: **-workspace <Directory-Pfad>** (optional)
  Definiert das Arbeitsdirectory des Clients. Der angegebene Pfad muss als Directory existieren. Ist diese Option nicht angegeben, wird das Arbeitsverzeichnis des eingeloggten Benutzers benutzt.

Alle weiteren Optionen des Clients erfahren Sie, wenn sie ihn mit der Option **-help** starten. Dann wird er nicht wirklich starten, sondern lediglich eine Hilfe-Information ausgeben.

## 3.2 Der Server Workspace

Ein Esprit-Server benötigt ein Arbeitsdirectory (seinen Workspace), wo er Daten speichern kann. Mit folgender Launch-Option kann ihm dieses zugewiesen werden:

- Option: **-workspace <Directory-Pfad>** (Pflicht)
  Definiert das Arbeitsdirectory des Servers. Der angegebene Pfad muss als Directory existieren.

In seinem Workspace-Verzeichnis legt der Server folgende Unterverzeichnisse an:

- **srvConfig**<
  hier befinden sich die Konfigurationsdateien des Servers
- **srvLogs**
  hier legt der Server seine Log-Dateien ab. Überflüssige Logdateien werden in regelmäßigen Abständen automatisch gelöscht, damit das Verzeichnis nicht unbegrenzt wächst.
- **srvData**
  hier legt der Server hoch geladene Dateien ab.
- **srvTask**
  hier speichert der Server die Konsolen-Ausgaben von asynchronen Tasks, die auf ihm gelaufen sind.
- **srvHistory**
  hier speichert der Server historische Daten von asynchronen Tasks, die auf ihm gelaufen sind

➔ **webData**
dies ist der Bereich, auf den der integrierte Webserver zugreift. Hier liegen typischerweise HTML Dokumente, die ein Client bei Bedarf abrufen kann.

Alle weiteren Optionen des Servers erfahren Sie, wenn sie ihn mit der Option **-help** starten. Dann wird er nicht wirklich hochfahren, sondern lediglich eine Hilfe-Information ausgegeben.

➔ Das beim Start des Servers angegebene Workspace-Directory kann leer sein. In diesem Falle erzeugt der Server selbst seine Konfigurationsdateien mit Default-Einträgen, die aber ggf. anzupassen sind.
Sollte eine Konfigurationsdatei fehlen, so erzeugt sie der Server beim Start automatisch neu mit Standardwerten.

## 3.3 Server-Konfigurationsdateien

Im Verzeichnis **srvConfig** speichert der Server seine Konfiguration. Verschiedene Aspekte der Serverkonfiguration sind in verschiedenen Dateien abgelegt. Die genaue Syntax der Einträge ist jeweils in den einzelnen Dateien beschrieben. Die Konfigurationsdateien im Einzelnen sind:

➔ **srvNetwork.ndf**
In dieser Datei sind alle im Netzwerk beteiligten Server mit Servername, Hostname und Port aufgeführt. Auch die Inter-Server-Verbindungen sind hier konfiguriert. Diese Datei muss auf allen Hosts, auf denen Esprit-Server im Verbund betrieben werden identisch sein.

➔ **srvStartup.ndf**
In dieser Datei befinden sich die statischen Startup-Konfigurationsparameter. Diese werden einmalig beim Hochfahren des Servers gelesen. Insbesondere ist hier der eindeutige Servername definiert, der auch in der *srvNetwork.ndf* Datei wiedergefunden werden muss. Diese Datei kann manuell mit Hilfe eines Texteditors geändert werden. Eine Beschreibung der einzelnen Parameter folgt weiter unten.

➔ **srvParams.ndf**
In dieser Datei befinden sich die dynamischen Startup-Konfigurationsparameter. Im Unterschied zur *srvStartup.ndf* Datei können die hier definierten Parameter sich während der Laufzeit des Servers ändern. In diesem Fall wird die *srvParams.ndf* Datei neu geschrieben. Änderungen hier sollten nicht manuell erfolgen, sondern über die Benutzeroberfläche des Admin-Clients. Eine Beschreibung der einzelnen Parameter folgt weiter unten.

➔ **srvUser.ndf**
Hier sind die Benutzer des Systems konfiguriert. Sie können über die Benutzeroberfläche des Admin-Clients verwaltet werden. Alternativ können Benutzer auch in einer Datenbank gespeichert werden.

➔ **srvGroup.ndf**
Hier sind die Benutzer-Gruppen des Systems konfiguriert. Sie können über die Benutzeroberfläche des Admin-Clients verwaltet werden. Alternativ können Gruppen auch in einer Datenbank gespeichert werden.

➔ **srvBatchTask.ndf**
In dieser Datei sind Java Klassen definiert, die als zeitgesteuerte Hintergrundprozesse laufen (ähnlich den Cron-Prozessen in Unix). Diese Prozesse laufen grundsätzlich nacheinander, immer nur einer zur gleichen Zeit. Die Zeit-Werte können über die Benutzeroberfläche des Admin-Clients eingestellt werden.

➔ **srvDataSource.ndf**
Die Einträge dieser Datei sind “benamte” Referenzen auf sog. Credential-Dateien, in denen die Parameter für eine Datenbank-Verbindung definiert sind. Die Referenzen können absolute oder relative Pfadnamen sein. Jeder Eintrag bedeutet eine zusätzliche Datenbank-Verbindung. Der Server kann beliebig viele Verbindungen auch zu Datenbanken unterschiedlicher Hersteller aufbauen. Die Dateien *dbDerbeEmb.cred* und *dbDerbyNet.cred* enthalten Beispiele für eine embedded- bzw. eine Netzwerk-Datenbankverbindung.

➔ **srvPermission.ndf**
Diese Datei beinhaltet Definitionen für Permissions, die angeben, auf welchem *PermissionControlled*-Objekt ein Benuter welche Aktionen ausführen darf. Sollte ein Benuter keine Rechte haben, so werden die Rechte der Gruppen überprüft, denen er angehört. Sollte auch dort keine Berechtigung gefunden werden, dann wird die betreffende Aktion vom Server abgelehnt.

## 3.4 Statische Server-Konfigurationsparameter

In der Datei **srvStartup.ndf** sind die statischen Konfigurationsparameter des Servers in Form von Key-Value Paaren definiert. Die wichtigsten Konfigurations-Keys im Einzelnen sind:

➔ **esprit.server.uniqueServerName**=EWB-Server
Definiert einen eindeutigen Namen für den Server. Dieser Name muss in *srvNetwork.ndf* wiedergefunden werden, wo die Netzwerkadresse festgelegt ist.

➔ **esprit.server.isManageUsersInDatabase**=false
Bestimmt, ob die Benutzer in der Datenbank (in den Tabellen *esprit_user*, *esprit_group*) oder in der Datei *srvUsers.ndf* bzw. *srvGroup.ndf* verwaltet werden.

➔ **esprit.server.isEmptyPasswordAllowed**=true
Bestimmt, ob ein Einloggen ohne Passwort erlaubt ist (z.B. für Gast-Benutzer)

➔ **esprit.server.hasWebService**=true
Definiert, ob der Web-Service gestartet wird. Nur wenn dieser konfiguriert ist, können HTML Dokumente abgerufen werden, bzw. Softwareaktualisierungen per Java-Web-Start durchgeführt werden.

➔ **esprit.webserver.webServicePort**=8080
Definiert den Netzwerk-Port an dem der Web-Service lauscht.

➔ **esprit.server.netIoType**= IO
Legt fest, ob der Server mit seinen Clients auf normale Art kommunizieren soll (IO) oder ob er alle Daten verschlüsselt empfangen und senden soll (IO_CRYPT).

➔ **esprit.server.isLogToConsole**=true
Definiert, ob der Server Logmeldungen auf die Systemkonsole schreibt. Grundsätzlich wird stets in eine Datei im Verzeichnis *srvLogs* geloggt.

➔ **esprit.server.resourceLockTimeout**=30
Setzt den Timeout Wert für Resource-Sperren. Wurde eine Sperre nicht innerhalb dieser Zeit getriggert, dann wird sie vom Server automatisch wieder freigegeben.

➔ **esprit.server.defaultAsyncAgentTimeout**=30
Setzt den Timeout Wert für die Ausführung von asynchronen Agenten. Hat der Agent nicht in dieser Zeit terminiert, wird dies als Fehlschlag angesehen.

➔ **esprit.server.messageSendQueueLength**=100
Definiert die Länge der Message-Send-Queue. In diesem Beispiel können sich 100 Messages im Sende-Kanal des Servers anhäufen bevor er das als Problem meldet.

- **esprit.server.requestRecvQueueLength**=100
  Definiert die Länge der Request-Empfangs-Queue. In diesem Beispiel können sich 100 Requests im Empfangs-Kanal des Servers anhäufen bevor er das als Problem meldet.
- **esprit.server.coServerReconnectDelay**=5
  Ein Esprit-Server kann mit anderen Esprit-Servern (sog. Co-Server) verbunden sein. Verliert er diese Verbindung, dann wird er im Rhythmus des hier konfigurierten Delays die neue Kontaktaufnahme versuchen.
- **esprit.email.mailserver**=my-mail-server
  In "Notfällen" sendet der Esprit-Server Emails an vorkonfigurierte Benutzer. Hier ist der Mail-Server definiert, an den diese Mails geschickt werden.
- **esprit.email.account.username**=my-username
  Definiert den Login-Benutzernamen für den Mail-Server
- **esprit.email.account.password**=my-password
  Definiert das Login-Passwort für den Mail-Server
- **esprit.email.sender**=my-sender-address
  Definiert welche Absender-Adresse in den Mails vom Server angegeben sein soll.
- **esprit.email.subject**=my-subject
  Definiert welcher Betreff in den Mails vom Server angegeben sein soll.

## 3.5 Dynamische Server-Konfigurationsparameter

In der Datei **srvParams.ndf** sind die dynamischen Konfigurationsparameter des Servers in Form von Properties (Key-Value Paaren) definiert. Die wichtigsten Properties im Einzelnen sind:

- **esprit.cpu.expectedMaxUsers**=100
  **esprit.cpu.maxParallelRequests**=50
  **esprit.cpu.maxParallelAgents**=50
  **esprit.cpu.maxParallelTransfers**=50
  Mit diesen Properties kann die maximale Anzahl von Threads gesteuert werden, die der Server für Benutzer, Requests, Agenten und Transfers verwendet. Requests, die diese Ressourcenbeschränkung überschreiten, werden vom Server abgelehnt.
- **esprit.server.log.logLevel**=INFO
  **esprit.server.log.maxLines**=3000
  **esprit.server.log.maxFiles**=5
  **esprit.server.log.isSendLogMessagesToClient**=true
  Diese Properties steuern die Logging-Parameter des Servers: Welches Log-Level verwendet wird, wie viele Zeilen in eine Log-Datei geschrieben werden bevor eine Neue angelegt wird, und wie viele Log-Dateien vorgehalten werden bevor Überflüssige gelöscht werden. Der letzte Key bestimmt, ob Server-Logmeldungen jeweils als Message zum Client geschickt werden. Falls dies erlaubt wird, kann z.B. ein Admin-Client die Server-Logmeldungen auf einer Client-Konsole beobachten.
- **esprit.server.encryptedDefaultUserPassword**=CWIrZ8enZZlK0oEfXhhsWA==
  Dies ist die Vorgabe eines Default-Passworts für neue Benutzer. Das hier gezeigte Passwort bedeutet im Klartext "*esprit*".
- **esprit.email.isSendEmailEnabled**=false
  Definiert, ob der Server im Problemfalle Emails zur Benachrichtigung schicken soll. Fall ja, muss ein Email-Server in der Konfigurationsdatei **servStartup.ndf** konfiguriert sein. Die Adressaten werden in der unten genannten Tabelle **EmailTarget** eingetragen.

➔ **EmailTarget**
In dieser Tabelle kann eine Liste von Email-Adressen angegeben werden. Alle genannten Empfänger erhalte System-Benachrichtigungen vom Server, falls er Probleme hat, oder sonstige Dinge melden möchte.