

IBM Security Directory Integrator
Versión 7.2.0.1

Guía de inicio



IBM Security Directory Integrator
Versión 7.2.0.1

Guía de inicio



Nota

Antes de utilizar esta información y el producto al que sirve de complemento, lea la información general contenida en la sección "Avisos" en la página 111.

Nota de edición

Nota: Esta edición es aplicable a la versión 7.2.0.1 del programa bajo licencia *IBM Security Directory Integrator* (5724-K74) y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

© Copyright IBM Corporation 2003, 2014.

Contenido

Figuras	v	Reglas de búsqueda = Criterios de enlace	66
		Descifrar errores de ejecución	67
Acerca de esta publicación	vii	Capítulo 3. Integración dirigida por	
Acceso a publicaciones y terminología	vii	sucesos	71
Accesibilidad	ix	Planificar líneas de ensamblaje	72
Formación técnica	ix	Líneas de ensamblaje de solicitud de servicio	73
Información de soporte	ix	Capítulo 4. Fortalecer las soluciones de	
Declaración de buenas prácticas de seguridad	ix	integración	81
Capítulo 1. Introducción	1	Legibilidad, reutilización y configurabilidad	81
Simplificar y resolver	3	Registro cronológico y auditoría	83
Arquitectura kernel/componente	4	Problemas de conectividad	83
Modelo de datos Entrada-Atributo-valor	5	Disponibilidad de las líneas de ensamblaje	84
Flujos de datos = Líneas de ensamblaje	6	Ampliación y rendimiento	86
Cómo empezar	8	Supervisión	87
		El depurador de líneas de ensamblaje	87
Capítulo 2. Información preliminar		Apéndice. Guía de EasyETL	89
sobre IBM Security Directory Integrator 11		Creación de un proyecto	92
Cómo crear su primera línea de ensamblaje	15	Detección de cambios	103
Ejecutar la línea de ensamblaje	30	Avisos	111
Comportamiento ante un valor nulo: Cuando faltan		Índice.	115
atributos o valores	32		
Depurar la línea de ensamblaje	40		
Buscar datos en un origen de datos secuencial.	48		
Utilización de la modalidad Buscar	59		
Herencia	63		

Figuras

1. El modelo de datos Entrada-Atributo-valor	6	40. Salida del registro con sus mensajes y el volcado de la entrada de trabajo	39
2. Flujo de datos a lo largo de una línea de ensamblaje	7	41. Depurar la línea de ensamblaje	40
3. Caso de ejemplo de aprendizaje	8	42. El repetidor de pasos de la línea de ensamblaje	41
4. Iniciar el Editor de configuración	11	43. Realización de pasos de la ejecución de LE	42
5. Seleccionar un espacio de trabajo	11	44. Realización de pasos del conector Write_XML_File	43
6. Pantalla de bienvenida del Editor de configuración	12	45. Modalidad de depurador avanzado	44
7. Dar un nombre al nuevo proyecto	13	46. Botones del depurador	45
8. Pantalla principal del Editor de configuración	14	47. Establecimiento de un punto de interrupción	46
9. Diagrama simplificado del caso de ejemplo con sólo dos orígenes de datos	15	48. Establecimiento de un punto de interrupción en un script	47
10. Recuadro de diálogo Nueva línea de ensamblaje	16	49. Línea de mandatos de evaluación JavaScript	48
11. Editor de una línea de ensamblaje vacía	17	50. Diagrama de flujo del caso de ejemplo	49
12. Añadir un nuevo componente	18	51. Arrastrar 'FullName' a la correlación de entrada del conector iterador	49
13. Elegir componente	19	52. Editar la asignación de 'FullName'	50
14. Renombrar el conector y cambiar la modalidad	20	53. Arrastrar el bucle de conector	51
15. Panel de configuración del Conector del sistema de archivos	21	54. Configuración del bucle de conector	51
16. Selección del analizador al añadir un nuevo objeto	22	55. Valores avanzados del bucle de conector	52
17. Selección del menú contextual Examinar datos	23	56. Atributos jerárquicos	52
18. El Explorador de datos	24	57. Arrastrar desde el Esquema hasta la Correlación de entrada	53
19. Descubrir el esquema de forma interactiva examinando los datos directamente	24	58. Editor de condiciones para la rama IF	53
20. LE con el conector iterador	25	59. Escribir el script para el enganche Fin de los datos	54
21. Botón Añadir componente	26	60. Lista de componentes de la sección Flujo de Datos de la línea de ensamblaje	55
22. LE con los dos conectores	27	61. Escribir un script de condición para la rama IF	56
23. Arrastrar los atributos a la Correlación de salida	28	62. Línea de ensamblaje completa con el bucle FOR-EACH	57
24. Renombrar una regla de correlación de atributos	28	63. Salida del registro con las estadísticas de la rama IF	58
25. Añadir el atributo 'FullName' a la correlación de salida	29	64. Salida XML con el atributo 'telephoneNo'	59
26. Editar la asignación	29	65. Función para copiar la línea de ensamblaje	60
27. El botón Ejecutar	30	66. Copiar una línea de ensamblaje al proyecto	61
28. Salida del registro de la ejecución de la línea de ensamblaje	30	67. Ejecutar la LE CreatePhoneDB	61
29. Barra de botones de la ventana Salida del registro	30	68. Salida del registro de la línea de ensamblaje 'CreatePhoneDB'	62
30. Examinar datos creados por un conector de salida	31	69. Arrastrar un conector a Recursos	63
31. Examinar el XML resultante	32	70. Arrastrar el nuevo recurso a la línea de ensamblaje	63
32. Botón Comportamiento ante un valor nulo para la configuración a nivel de la LE	33	71. Establecer la herencia para la pestaña enganches	64
33. Diálogo de comportamiento del Comportamiento ante un valor nulo	34	72. Restaurar la herencia en una regla de correlación	65
34. Resultado en la salida XML de los valores de Comportamiento ante un valor nulo	34	73. Cambiar la modalidad, descubrir y correlacionar atributos	65
35. Seleccionar el componente de rama IF	35	74. Un criterio de enlace simple	66
36. Editar condiciones para la rama IF	36	75. Mensaje de error en la salida del registro	67
37. Añadir condiciones simples a la rama IF	37	76. Diagrama de flujo parcial de la modalidad Buscar	68
38. Su primera línea de ensamblaje completa	38	77. Primer ejercicio de la guía de aprendizaje completado	69
39. Restablecer el Comportamiento ante un valor nulo de la línea de ensamblaje	39	78. Planificador de IBM Security Directory Integrator	73

79. Panel de correlación de atributos del Conector de servidor HTTP	74	93. Prueba de la conexión y descubrimiento del esquema	95
80. Añadir un elemento de correlación atributos de entrada	75	94. Origen de entrada configurado	96
81. Elemento de correlación comodín	76	95. Renombrar un atributo de salida	97
82. Propiedades de cabecera TCP y HTTP devueltas como atributos	77	96. Un registro leído y recopilado	98
83. Arrastrar el componente de función de línea de ensamblaje.	78	97. Línea de ensamblaje EasyETL completada	98
84. Volcado de la entrada de trabajo seguido de las estadísticas de la LE	79	98. Habilitación de la transformación	99
85. Línea de ensamblaje TINA_WebServer completada	80	99. Mostrar script de transformación	100
86. Interfaz web simple a la solución	80	100. Evaluar expresión	100
87. Pantalla de bienvenida.	90	101. Recopilación de salida con el atributo FullName calculado	101
88. Figura 2. Entorno de trabajo de EasyETL	91	102. Salida XML	102
89. Botón Nuevo proyecto	92	103. Configuración de delta	103
90. Editor de línea de ensamblaje simple	92	104. Todas las entradas sin modificar y omitidas	104
91. Selección de información de origen	93	105. Selección del criterio de enlace.	105
92. Establecimiento del parámetro Vía de acceso del archivo.	94	106. Creación de activos de línea de mandatos para ejecutar el trabajo ETL.	106
		107. Ejecución del trabajo ETL a la máxima velocidad	107
		108. Definición de criterios de enlace para un conector de entrada	108

Acerca de esta publicación

Utilice la IBM® Security Directory Integrator Guía de inicio para descubrir los Security Directory Integrator conceptos básicos y comprender cómo diseñar soluciones de integración de datos de forma efectiva.

Este documento presenta información conceptual acerca de IBM Security Directory Integrator y proporciona ejemplos para ayudarle a iniciarse con el producto.

Acceso a publicaciones y terminología

Lea las descripciones de la biblioteca de IBM SecurityDirectory Integrator Versión 7.2.0.1 y las publicaciones relacionadas que puede acceder en línea.

Esta sección proporciona:

- Una lista de publicaciones contenidas en la “Biblioteca de IBM SecurityDirectory Integrator”.
- Enlaces a “Publicaciones en línea” en la página viii.
- Un enlace al “Sitio web de terminología de IBM” en la página ix.

Biblioteca de IBM SecurityDirectory Integrator

Los documentos siguientes están disponibles en la biblioteca de IBM SecurityDirectory Integrator:

- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Federated Directory Server Administration Guide*
Contiene información sobre la utilización de la consola de Federated Directory Server para diseñar, implementar y administrar soluciones de integración de datos. También contiene información sobre cómo utilizar el protocolo SCIM (System Cross-Domain Identity Management) y la interfaz de gestión de identidades.
- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Guía de inicio*
Contiene una breve guía de aprendizaje e información básica sobre IBM SecurityDirectory Integrator. Incluye ejemplos para crear interacción y obtener experiencia práctica sobre IBM SecurityDirectory Integrator.
- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Guía del usuario*
Contiene información sobre el uso de IBM SecurityDirectory Integrator. Contiene instrucciones para diseñar soluciones utilizando la herramienta de diseño de SecurityDirectory Integrator (Editor de configuración) o para ejecutar soluciones ya preparadas desde la línea de mandatos. También proporciona información sobre interfaces, conceptos y la creación desde AssemblyLine.
- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Guía de instalación y del administrador*
Incluye información completa sobre cómo instalar, migrar de una versión anterior, configurar la función de registro y el modelo de seguridad subyacente de la API de servidor remoto de IBM SecurityDirectory Integrator. Contiene información sobre cómo desplegar y gestionar soluciones.
- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Guía de referencia*

Contiene información detallada acerca de los componentes individuales de IBM SecurityDirectory Integrator: conectores, componentes de función, analizadores, objetos, etc., que son los bloques de creación de la línea de ensamblaje.

- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Guía de determinación de problemas*
Proporciona información acerca de herramientas, recursos y técnicas de IBM SecurityDirectory Integrator que pueden resultar de ayuda en la identificación y resolución de problemas.
- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Guía de mensajes*
Proporciona una lista de todos los mensajes informativos, de advertencia y de error asociados con IBM SecurityDirectory Integrator.
- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Guía de conectores de sincronización de contraseñas*
Incluye información completa para instalar y configurar cada uno de los cinco plug-ins de sincronización de contraseñas de IBM: Windows Password Synchronizer, Sun Directory Server Password Synchronizer, IBM Security Directory Server Password Synchronizer, Domino Password Synchronizer y Password Synchronizer para UNIX y Linux. También proporciona instrucciones de configuración para los almacenes de contraseñas LDAP y JMS.
- *IBM SecurityDirectory Integrator Versión 7.2.0.1 Release Notes*
Describe las características nuevas y la información de última hora acerca de IBM SecurityDirectory Integrator que no se ha podido incluir en la documentación.

Publicaciones en línea

IBM publica publicaciones sobre un producto en las ubicaciones siguientes cuando se comercializa el producto y cuando se actualizan las publicaciones:

Biblioteca de IBM SecurityDirectory Integrator

El sitio de documentación del producto (<http://www-01.ibm.com/support/knowledgecenter/SSCQGF/welcome>) muestra la página de bienvenida y de navegación para esta biblioteca.

IBM Security Systems Documentation Central

IBM Security Systems Documentation Central proporciona una lista alfabética de todas las bibliotecas de productos de IBM Security Systems y enlaces a la documentación en línea correspondiente a versiones específicas de cada producto.

IBM Publications Center

El sitio IBM Publications Center (<http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>) ofrece funciones de búsqueda personalizada para ayudarle a encontrar todas las publicaciones de IBM que necesite.

Información relacionada

La información relacionada con IBM SecurityDirectory Integrator está disponible en las siguientes ubicaciones:

- IBM SecurityDirectory Integrator utiliza el cliente JNDI de Oracle. Para obtener información sobre el cliente JNDI, consulte la publicación *Java Naming and Directory Interface™ Specification* en <http://download.oracle.com/javase/7/docs/technotes/guides/jndi/index.html>.

- La información que le puede ayudar a responder sus preguntas sobre IBM SecurityDirectory Integrator se encuentra en https://www.ibm.com/support/entry/myportal/over-accesspubsview/software/security_systems/tivoli_directory_integrator.

Sitio web de terminología de IBM

El sitio web de terminología de IBM agrupa terminología para bibliotecas de productos en una sola ubicación. Puede acceder al sitio web de terminología en <http://www.ibm.com/software/globalization/terminology>.

Accesibilidad

Las funciones de accesibilidad ayudan a los usuarios con alguna discapacidad física, tal como una movilidad o visión restringida, a utilizar satisfactoriamente los productos de software. Con este producto, puede utilizar tecnologías de asistencia a discapacitados para oír o navegar por la interfaz. También puede utilizar el teclado en lugar del ratón para ejecutar todas las funciones de la interfaz gráfica de usuario.

Para obtener información adicional, consulte el apéndice sobre accesibilidad de *Configuración de Directory Integrator*.

Formación técnica

Para obtener información sobre formación técnica, consulte el siguiente sitio web de IBM Education en <http://www.ibm.com/software/tivoli/education>.

Información de soporte

Soporte de IBM proporciona asistencia para problemas relacionados con código fuente y preguntas habituales de corta duración sobre instalación o uso. Puede acceder directamente al sitio IBM Software Support en <http://www.ibm.com/software/support/probsub.html>.

Resolución de problemas proporciona detalles sobre:

- Qué información recopilar antes de ponerse en contacto con Soporte de IBM.
- Los distintos métodos para ponerse en contacto con Soporte de IBM.
- Cómo utilizar IBM Support Assistant.
- Instrucciones y recursos de determinación de problemas para identificar y corregir problemas sin ayuda exterior.

Declaración de buenas prácticas de seguridad

La seguridad de los sistemas de tecnología de la información implica proteger los sistemas y la información mediante la prevención, detección y respuesta al acceso no autorizado desde dentro y fuera de su empresa. El acceso no autorizado puede dar como resultado la alteración, destrucción, apropiación indebida o mal uso de la información, y también daños en los sistemas o mal uso de ellos, incluida su utilización para atacar a otros sistemas. Ningún producto o sistema de la tecnología de la información se debe considerar completamente seguro y ningún producto, servicio o medida de seguridad puede ser completamente efectivo para prevenir la utilización o acceso indebidos. Los sistemas, productos y servicios de IBM están diseñados para formar parte de un sistema de seguridad completo, que necesariamente incluye procedimientos operativos adicionales y puede necesitar

otros sistemas, productos o servicios para lograr la máxima efectividad. IBM DOES NO GARANTIZA QUE NINGÚN SISTEMA, PRODUCTO O SERVICIO SEA INMUNE, O HAGAN A SU EMPRESA INMUNE A LA CONDUCTA MALINTENCIONADA O ILEGAL DE CUALQUIERA DE LAS PARTES.

Capítulo 1. Introducción

Este manual es una introducción sencilla a un sistema sencillo. No se equivoque; la palabra *sencillo* se utiliza aquí en su contexto más positivo e intenso, porque la mejor forma de enfrentarse a un problema complejo es simplificarlo; dividirlo en partes más manejables y aprender a dominar esas partes constituyentes. Divida y vencerá. Esta técnica, que se utiliza instintivamente para resolver los problemas diarios, es igualmente aplicable para el intercambio de información técnica en una oficina, en una empresa o en todo el globo.

Si tiene ganas de empezar a extraer información rápidamente de archivos, directorios, bases de datos o Lotus Notes y de transferir estos datos a otro lugar, puede saltar directamente al apéndice, “Guía de EasyETL”, en la página 89. Esta característica le permite aprovechar la eficacia de IBM Security Directory Integrator sin tener que aprender primero los conceptos clave. En lugar de eso, puede elegir el origen y el destino y después pulsar Ejecutar y observar el flujo de datos. Si, por otro lado, desea más control sobre cómo leer, filtrar, enriquecer, transformar y mover los datos, siga leyendo este capítulo; puede leer “Guía de EasyETL”, en la página 89 más adelante.

IBM Security Directory Integrator¹ está diseñado y construido bajo la premisa de que incluso los problemas más complejos de integración se pueden descomponer en tres componentes básicos:

- Los sistemas que participan en la comunicación – también llamados *orígenes de datos*,
- Los *flujos de datos* entre estos sistemas,
- Los *sucesos* que desencadenan los flujos de datos.

Con IBM Security Directory Integrator puede traducir esta comprensión de las partes de un problema de integración directamente en una solución, construyéndola de forma incremental, de flujo a flujo, comprobando y verificando continuamente su solución. Con esta forma de trabajar, resulta más fácil estimar y planificar los proyectos de integración, a menudo reduciéndose a contar y presupuestar los flujos de datos a implementar. El hecho de realizar una tarea en pasos ejecutables, le permite también mostrar los avances a las personas implicadas.

IBM Security Directory Integrator también acelera el desarrollo, abstrayendo las diferencias técnicas entre sus orígenes de datos, lo que le permite destinar más tiempo a concentrarse en los requisitos empresariales.

IBM Security Directory Integrator aprovecha los recursos de Eclipse y su entorno de desarrollo es a la vez completo y ampliable. Mediante los proyectos de integración se crean bibliotecas de componentes y lógica empresarial que pueden ser reutilizados inmediatamente para solucionar nuevos problemas. De esta forma, los equipos de trabajo de toda la organización pueden compartir los activos de IBM Security Directory Integrator, obteniendo como resultado proyectos independientes – incluso soluciones puntuales – que encajan inmediatamente en una infraestructura integrada y gestionada coherentemente.

1. No permita que el nombre le confunda; IBM Security Directory Integrator no está limitado al trabajo con directorios, permite trabajar con la mayoría de almacenes de datos, transportes, protocolos y APIs – incluidos, por supuesto, los directorios LDAP.

Este documento le ofrece una introducción a esta metodología consistente en simplificar los problemas para solucionarlos descrita más arriba. También aprenderá a aprovechar la elegante simplicidad del conjunto de herramientas de IBM Security Directory Integrator. Concretamente, de los dos programas siguientes:

- El entorno de desarrollo, denominado *Editor de configuración*, también abreviado como 'CE'.
- El motor de tiempo de ejecución, denominado simplemente el *Servidor*.

Crearé sus soluciones de IBM Security Directory Integrator con el CE, mientras se utilizan uno o más servidores para ejecutarlas. Estos programas trabajan conjuntamente, haciendo que la experiencia del usuario sea fluida e incluso permitiendo trabajar en distintas plataformas; por ejemplo, desarrollando las soluciones en el portátil y ejecutándolas de forma remota en un sistema principal para probarlas y depurarlas.

Escribir scripts en JavaScript

Como se ha mencionado anteriormente, IBM Security Directory Integrator le permite crear rápidamente soluciones de integración. Sin embargo, para ampliar la funcionalidad automática que viene incorporada en el producto con sus propios procesos y su comportamiento de flujos personalizado, tendrá que escribir fragmentos de código script.

La escritura de scripts se realiza en JavaScript, y IBM Security Directory Integrator incluye el motor IBM JSEngine para proporcionar un entorno de escritura de scripts rápido y fiable. Esto significa que tendrá que conocer y utilizar los fundamentos del lenguaje JavaScript. Dispone de diversos recursos en línea e impresos para aprender JavaScript. Consulte los grupos de noticias y los sitios web de IBM Security Directory Integrator para obtener recomendaciones y enlaces.

Para obtener más información sobre la escritura de scripts en IBM Security Directory Integrator, consulte la publicación *Configuración de Directory Integrator*.

Instalación de IBM Security Directory Integrator

IBM Security Directory Integrator se instala en unos minutos y, una vez instalado, puede empezar inmediatamente a crear, probar y desplegar soluciones. Funciona en una amplia variedad de plataformas, entre las que se incluyen Microsoft Windows, IBM AIX, IBM System z, y diversos entornos de UNIX y Linux.

Hay tres vías de acceso importantes al instalar IBM Security Directory Integrator, y el programa de instalación le pedirá que especifique las dos primeras:

1. El *Directorio de instalación*, donde se guardan los archivos de programa, junto con los archivos de proceso por lotes o los scripts utilizados para lanzar las diversas herramientas.
2. El *Directorio de soluciones*, a menudo abreviado 'SolDir', que es la carpeta que está activa cuando ejecuta IBM Security Directory Integrator. Observará que archivos de proceso por lotes y scripts de inicio del entorno de desarrollo del Editor de configuración (ibmditk) y del Servidor (ibmdisrv) empiezan con un mandato para cambiar al Directorio de soluciones. En consecuencia, todas las vías de acceso relativas que utilice en su solución partirán de su Directorio de soluciones.

3. La carpeta del *espacio de trabajo*. Es donde se guardan los archivos de proyecto y de recursos². Será, por defecto, una carpeta denominada "workspace" en su Directorio de soluciones.

Para obtener más información sobre la instalación de IBM Security Directory Integrator, consulte las *Instrucciones de instalación de IBM Security Directory Integrator en Instalación y administración*.

Instalación de los archivos de la guía de aprendizaje

Los ejercicios de aprendizaje que se explican en este libro necesitan archivos de datos de soporte que se encuentran en la subcarpeta `examples/Tutorial` del directorio de instalación de IBM Security Directory Integrator. Por ejemplo, una instalación de Windows estándar pondría estos archivos en el directorio siguiente:

```
C:\Program Files\IBM\TDI\V7.2\examples\Tutorial
```

El directorio 'Tutorial' debe contener los siguientes archivos:

- `CreatePhoneDB.assemblyline`
- `index.html`
- `OtherPage.html`
- `People.csv`
- `PhoneNumbers.xml`
- `readme.txt`
- `Return web page.script`

Nota: Como se ha indicado en el apartado anterior, el instalador la solicitará que especifique la ubicación de su Directorio de soluciones. En este directorio se almacenarán los archivos de proyectos y recursos. Normalmente será un subdirectorio denominado `My Documents\TDI` contenido en el área de inicio.

Copie la carpeta `Tutorial` a su Directorio de soluciones para que esté más rápidamente accesible desde el Editor de configuración.

Simplificar y resolver

Esta sección le ayudará a comprender el punto de partida al diseñar una solución de integración de datos. Aunque la estrategia de diseño es incremental, es adecuada para cualquier proyecto de integración y despliegue de sistemas independientemente del tamaño, incluso para proyectos muy grandes.

Utilice una estrategia como la siguiente para diseñar paso a paso una solución de integración de datos utilizando IBM Security Directory Integrator:

- Reduzca la complejidad dividiendo el problema en partes más pequeñas y manejables.
- Empiece con una porción de la solución global, preferiblemente una porción que se pueda completar en una o dos semanas.
- Empiece con una porción de la solución global que se pueda poner en producción de forma individual.

2. Estos términos se explican en Capítulo 2, "Información preliminar sobre IBM Security Directory Integrator", en la página 11

Manejo de proyectos complejos

El mejor método para simplificar proyectos complejos de integración y despliegue de sistemas es reducir la complejidad mediante el desglose del problema en componentes más pequeños y manejables. Una vez hecho esto, empiece a trabajar en una porción de la solución global, preferiblemente una porción que se pueda desplegar de forma independiente. De esta forma, estará ya obteniendo un retorno de la inversión mientras aborda el resto.

Tras aislar la parte con la que va a trabajar, simplifíquela más centrándose en las unidades básicas de comunicación: los propios flujos de datos. Ahora está en disposición de iniciar la implementación. El desarrollo de la integración se efectúa con el Editor de configuración de IBM Security Directory Integrator (abreviado 'CE') a través de una serie de ciclos de intentar-probar-perfeccionar, lo que hace que el proceso sea iterativo e incluso de exploración. Esto no sólo le ayuda a saber más sobre su propia instalación, sino que también le permite hacer evolucionar su solución de integración a medida que aumenta su conocimiento del conjunto de problemas y el impacto sobre su infraestructura.

Temas relacionados

Consulte los temas siguientes para ver una explicación de cómo IBM Security Directory Integrator le permite transformar datos utilizando líneas de ensamblaje.

- “Arquitectura kernel/componente”
- “Modelo de datos Entrada-Atributo-valor” en la página 5
- “Flujos de datos = Líneas de ensamblaje” en la página 6

Arquitectura kernel/componente

Una cualidad fundamental de IBM Security Directory Integrator es su diseño de kernel/componente.

El término *kernel* aquí hace referencia a la infraestructura RID (Rapid Integration Development) que le permite crear rápidamente sus soluciones de integración y proporciona lógica de ejecución automática para conducirlos. Características, que de otro modo tendría que codificar a mano (y por esto a menudo se desestiman) como módulos de registro/rastreo, recuperación de conexión, detección de cambios, manejo de errores y una API de gestión externa, están inmediatamente disponibles incluso para el flujo de datos más simple.

Además de esta funcionalidad genérica de kernel, IBM Security Directory Integrator proporciona un conjunto de componentes específicos del origen de datos: objetos auxiliares que hacen que el usuario no necesite conocer los detalles técnicos de la interacción con orígenes de datos. Los dos tipos de componentes que utilizará más son los *Conectores* y los *Analizadores*.

Los conectores proporcionan conectividad a una amplia variedad de orígenes de datos y, además, manejan inherentemente datos estructurados con independencia de su organización subyacente. Algunos conectores también hacen la función de manejadores de sucesos, por ejemplo, enlazar con puertos IP y esperar conexiones entrantes, o 'estar a la escucha' de cambios que tengan lugar en los directorios, bases de datos o archivos.

Por otro lado, los analizadores se utilizan para tratar datos no estructurados – es decir, secuencias de bytes, como los que se encuentran en archivos, correos electrónicos POP3/SMTP, mensajes MQ y secuencias de datos que pasen por puertos IP.

IBM Security Directory Integrator proporciona una biblioteca ampliable de Conectores y Analizadores, cada uno de ellos diseñado para trabajar con un sistema, servicio, API, transporte o formato específico. La naturaleza intercambiable de los componentes de IBM Security Directory Integrator le permite crear una solución basándose en datos de prueba – por ejemplo, archivos de texto – y después simplemente cambiar los Conectores utilizados para que la solución apunte a orígenes reales para su verificación y despliegue.

Además, los componentes de IBM Security Directory Integrator son fáciles de usar, de crear y de ampliar. Puede aumentar su biblioteca para tratar orígenes de datos y servicios personalizados descargando nuevos componentes de un sitio web comunitario, escribiendo sus propios componentes en Java™, o creándolos y probándolos de forma interactiva utilizando scripts directamente en el CE.

Modelo de datos Entrada-Atributo-valor

La forma en que se organizan y se almacenan los datos difiere mucho de sistema a sistema:

- Las bases de datos almacenan información en filas, normalmente con un número fijo de columnas, de las cuales cada una contiene un único valor para un registro dado;
- Los directorios mantienen entradas orientadas a objetos que contienen un número variable de atributos. Éstos a su vez tienen cero, uno o diversos valores³;
- Las bases de datos de Lotus Domino contienen Documentos que están compuestos de Campos, cada uno de los cuales puede tener un valor o varios.
- Otros sistemas representan sus contenidos de datos en forma de nodos, objetos, registros secuencias de bytes con formato o conjuntos de pares clave-valor.

Para que la comunicación tenga sentido para todos los participantes, los formatos de datos deben ser compatibles o se deben traducir para adaptarlos a cada sistema. Esto se llama *empaquetamiento de datos* y es a menudo el primer obstáculo que debe abordar un especialista en integración – y superarlo puede consumir rápidamente una parte considerable de los recursos del proyecto. Los conectores de IBM Security Directory Integrator solucionan este problema convirtiendo automáticamente los tipos específicos de un origen determinado en una representación canónica consistente. Los valores de datos individuales se traducen a objetos Java relevantes, y los tipos nativos comparables se representan de la misma forma. Por ejemplo, las líneas que se leen de archivos, los atributos de serie LDAP, los campos de texto Domino y las columnas RDBMS CHAR y VARCHAR son convertidas a `java.lang.String` por sus conectores respectivos.

Estos valores ordenados se acumulan después en Atributos: objetos Java especializados definidos por IBM Security Directory Integrator. Como se ha indicado arriba, algunos orígenes sólo permiten un único valor por columna o campo, mientras que otros permiten diversos valores bajo el mismo nombre de atributo. Un atributo de IBM Security Directory Integrator admite tanto

3. Por ejemplo, usted probablemente tiene diversas direcciones de correo electrónico y se pueden almacenar todas ellas en el atributo multivalor titulado 'correo' en el directorio de empleados de su empresa

implementaciones de valor único como de diversos valores y puede incluso no tener ningún valor si es necesario, por ejemplo para representar una columna nullable en una base de datos.

Todos los atributos que constituyen una unidad de datos (es decir, un registro, un mensaje, un documento, etc.) se recopilan en otro objeto de IBM Security Directory Integrator denominado una *Entrada*. Una entrada puede contener cualquier número de atributos o ninguno.

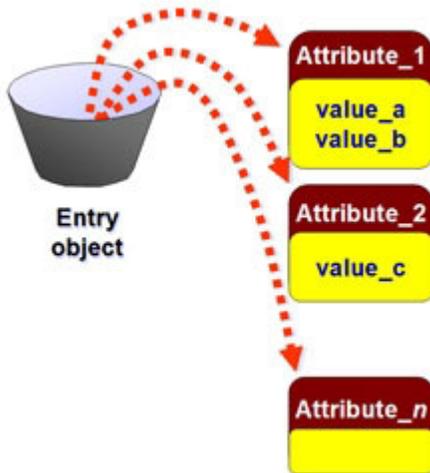


Figura 1. El modelo de datos Entrada-Atributo-valor

Cada flujo de datos tiene un 'contenedor' o entrada principal denominado Entrada de Trabajo. Cuando un conector lee datos de entrada, crea atributos y los pone en la entrada de trabajo. Un conector cualquiera configurado como salida utiliza los atributos que se encuentran en la entrada de trabajo para provocar cambios en los sistemas de destino.

Esta forma de trabajar, en dos etapas, proporciona una flexibilidad casi ilimitada en la forma como se transfieren, se transforman, se filtran y se enriquecen los datos. También significa que primero puede construir su flujo de datos completo con conectores de entrada y después examinar los datos de forma interactiva con el CE, a medida que se leen y manipulan, sin necesidad de pensar en las conexiones con los sistemas de salida.

Como se verá más adelante, una entrada de IBM Security Directory Integrator maneja datos jerárquicos complejos con la misma facilidad con que maneja esquemas planos.

Flujos de datos = Líneas de ensamblaje

Cada flujo de datos de una solución se implementa en forma de IBM Security Directory Integrator *línea de ensamblaje*, también abreviada 'LE' en este manual y en otra bibliografía existente.

Las LE son listas ordenadas de componentes que conforman una única ruta continua desde los orígenes de entrada hasta los destinos. El comportamiento predefinido proporcionado por el kernel relaciona los componentes entre sí y pasa datos almacenados en la Entrada de trabajo de un componente al siguiente.

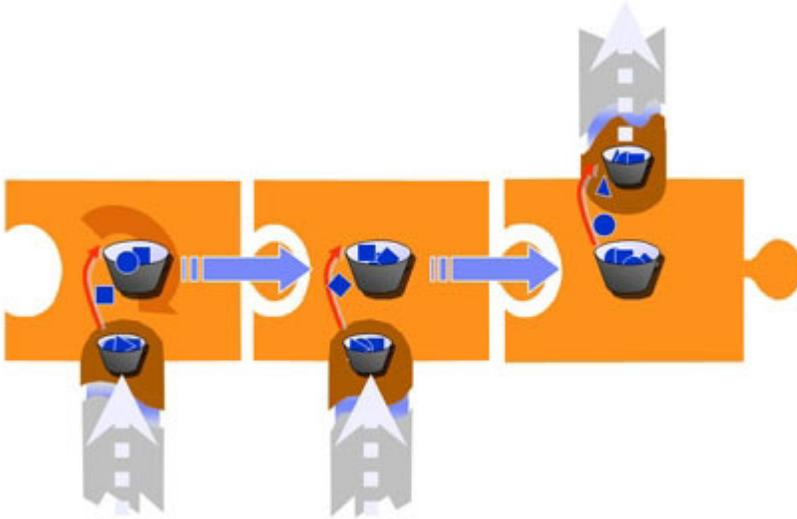


Figura 2. Flujo de datos a lo largo de una línea de ensamblaje

Se dice que una imagen vale más que mil palabras y este diagrama no es una excepción. Las tres piezas de puzzle representan Conectores, enlazados para formar una Línea de ensamblaje. La parte más oscura de cada pieza de puzzle representa la parte del Conector específica del origen de datos – es decir, la *interfaz* con el sistema conectado – conocida como la *Interfaz de Conector* (abreviada 'CI'). La parte restante, más clara, de cada pieza de puzzle representa la funcionalidad genérica del kernel que hace que todos los componentes funcionen de una forma similar y predecible, permitiendo enlazarlos y proporcionando patrones automatizados de comportamiento con puntos de control para su personalización⁴.

La imagen ilustra también otros conceptos importantes. Por ejemplo, además de la Entrada de trabajo que se muestra en la imagen pasando de componente a componente a lo largo de la línea de ensamblaje, hay un "contenedor" Java adicional en cada una de las interfaces de conector. Cada objeto de Entrada local se utiliza para guardar datos en memoria caché durante las operaciones de lectura y escritura que realiza ese CI, y se denomina su *Entrada de conector*.

Observe ahora las flechas que ilustran el flujo de datos entre las diversas Entradas de conector y la Entrada de trabajo de la LE. Son las *Correlaciones de atributos* y cada una representa un conjunto de reglas para el movimiento y transformación de datos en su paso hacia dentro o hacia fuera de la línea de ensamblaje. Los elementos que transportan datos de una Entrada de conector a una Entrada de trabajo se denominan *Correlaciones de entrada* porque determinan los datos que entran en la Línea de ensamblaje. La flecha de la pieza de puzzle que está más a la derecha y que muestra datos que fluyen en la dirección opuesta - de la entrada de trabajo a la entrada de conector – se denomina *Correlación de salida*.

Puesto que sólo hay una entrada de trabajo a la vez, puede deducir que las líneas de ensamblaje procesan un elemento a la vez: por ejemplo, una fila de base de datos, una entrada de directorio, un mensaje MQ, etcétera. Este es otro aspecto importante de IBM Security Directory Integrator y, aunque por una línea de

4. Como se puede observar, cada componente de línea de ensamblaje refleja la arquitectura de kernel/componente de IBM Security Directory Integrator. Si decide crear su propio componente, sólo tendrá que implementar su interfaz. La "envoltura", en forma de línea de ensamblaje, y la funcionalidad que tiene incorporada están disponibles automáticamente, por 'gentileza' del kernel de IBM Security Directory Integrator.

ensamblaje pueden pasar cientos e incluso miles de Entradas por segundo⁵, ésta es una consideración importante al diseñar su solución. Por supuesto, también se puede distribuir el trabajo entre diversas líneas de ensamblaje; encontrará ésta y otras técnicas para optimizar el rendimiento de las LE en la bibliografía de IBM Security Directory Integrator.

Cómo empezar

Una buena forma de iniciar un proyecto de integración es hacer un diagrama del problema que se tiene entre manos.

Con un lápiz y un trozo de papel, trace los flujos deseados a grandes rasgos. Este ejercicio no sólo le ayuda a visualizar el alcance de la tarea, también sirve como anteproyecto para la implementación de estos flujos en IBM Security Directory Integrator.

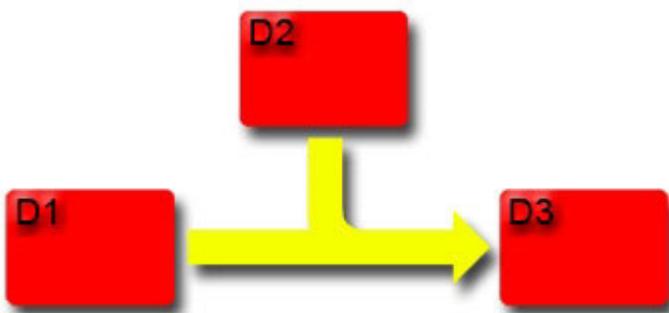


Figura 3. Caso de ejemplo de aprendizaje

El primer paso para crear una solución de IBM Security Directory Integrator es convertir los flujos de datos entre orígenes de datos en líneas de ensamblaje formadas por conectores. El paradigma de IBM Security Directory Integrator, 'simplificar y resolver', prescribe crear la solución de forma incremental, empezando de la forma más sencilla posible.

Para ilustrar esta idea, veremos el caso de ejemplo que utilizará para su primera línea de ensamblaje. En esta tarea de integración participan tres orígenes de datos, etiquetados como D1, D2 y D3. El objetivo es crear una solución que migre los contenidos de D1 a D3, aumentando estos datos con valores que se encuentran en D2. Al trasladar este requisito a una línea de ensamblaje, obtenemos tres conectores, uno para cada origen de datos:

1. el primer conector deberá *iterar* en D1, alimentando el flujo con estos datos;
2. el segundo conector *buscará* registros relacionados en D2 y mezclará estos valores con los valores procedentes de D1;
3. finalmente, tendremos un tercer conector configurado para *añadir* estos registros aumentados a D3.

En vez de abordar todo el problema de una vez, IBM Security Directory Integrator le permite simplificar la tarea empezando con sólo dos conectores: uno que lea el contenido de D1 y lo pase a la LE y otro que escriba estos valores en D3. Una vez esta línea de ensamblaje reducida funcione correctamente, se podrá ampliar con el conector para D2 para añadir los atributos adicionales. Ésta es precisamente la

5. El rendimiento dependerá del diseño y la complejidad de la línea de ensamblaje y de la configuración de la máquina donde se ejecute el servidor.

forma como creará su primera solución de IBM Security Directory Integrator, y los pasos para guiarle en este proceso constituyen el resto de esta guía.

Capítulo 2. Información preliminar sobre IBM Security Directory Integrator

Esta sección ofrece información útil para comprender los aspectos básicos de IBM Security Directory Integrator, así como un conjunto de ejercicios de aprendizaje para adquirir experiencia práctica en el uso del entorno de desarrollo.

El primer paso para empezar a conocer el producto es iniciar la herramienta de desarrollo de IBM Security Directory Integrator, conocida como el Editor de configuración o CE.

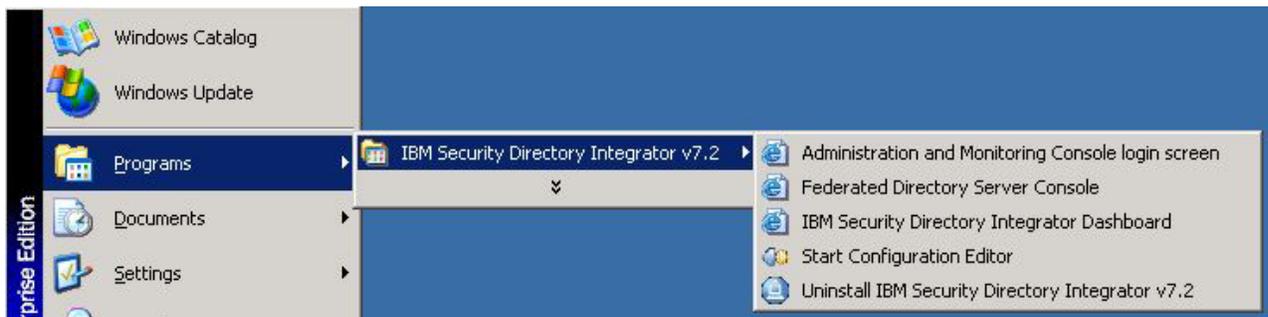


Figura 4. Iniciar el Editor de configuración

La primera vez que inicie el CE, obtendrá este diálogo donde podrá especificar el espacio de trabajo.

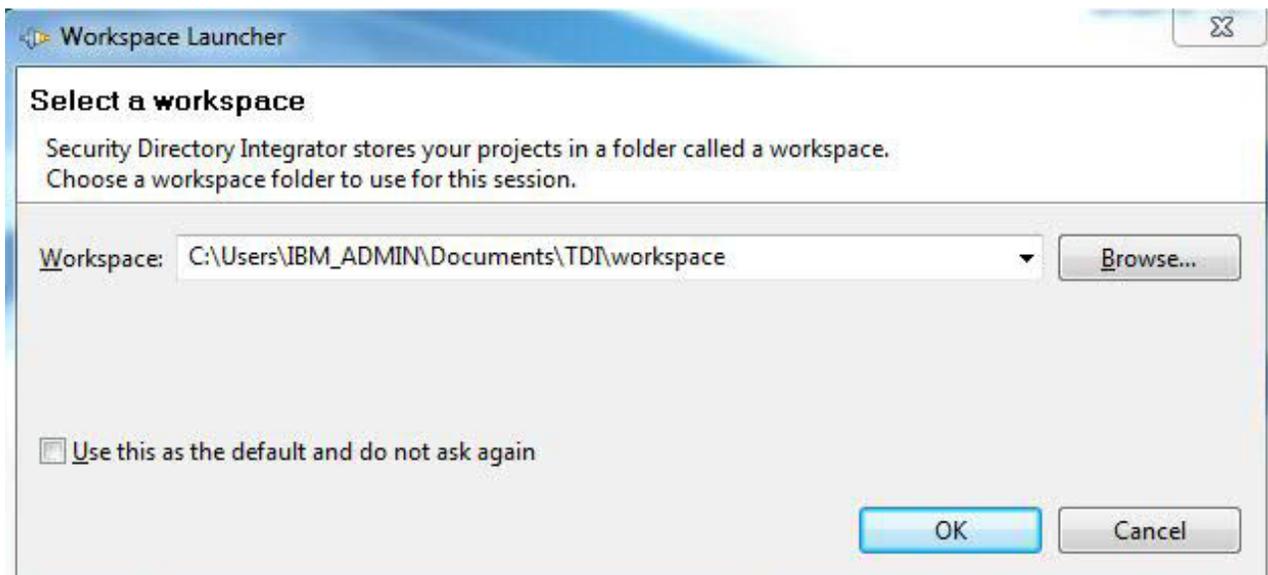


Figura 5. Seleccionar un espacio de trabajo

El espacio de trabajo es el lugar donde el Editor de configuración almacenará los archivos de proyecto, incluyendo componentes y líneas de ensamblaje. Normalmente se encuentra bajo el Directorio de soluciones.

Una vez esté satisfecho con la ubicación del espacio de trabajo, pulse el botón **Aceptar**. A continuación, aparecerá la pantalla de bienvenida.

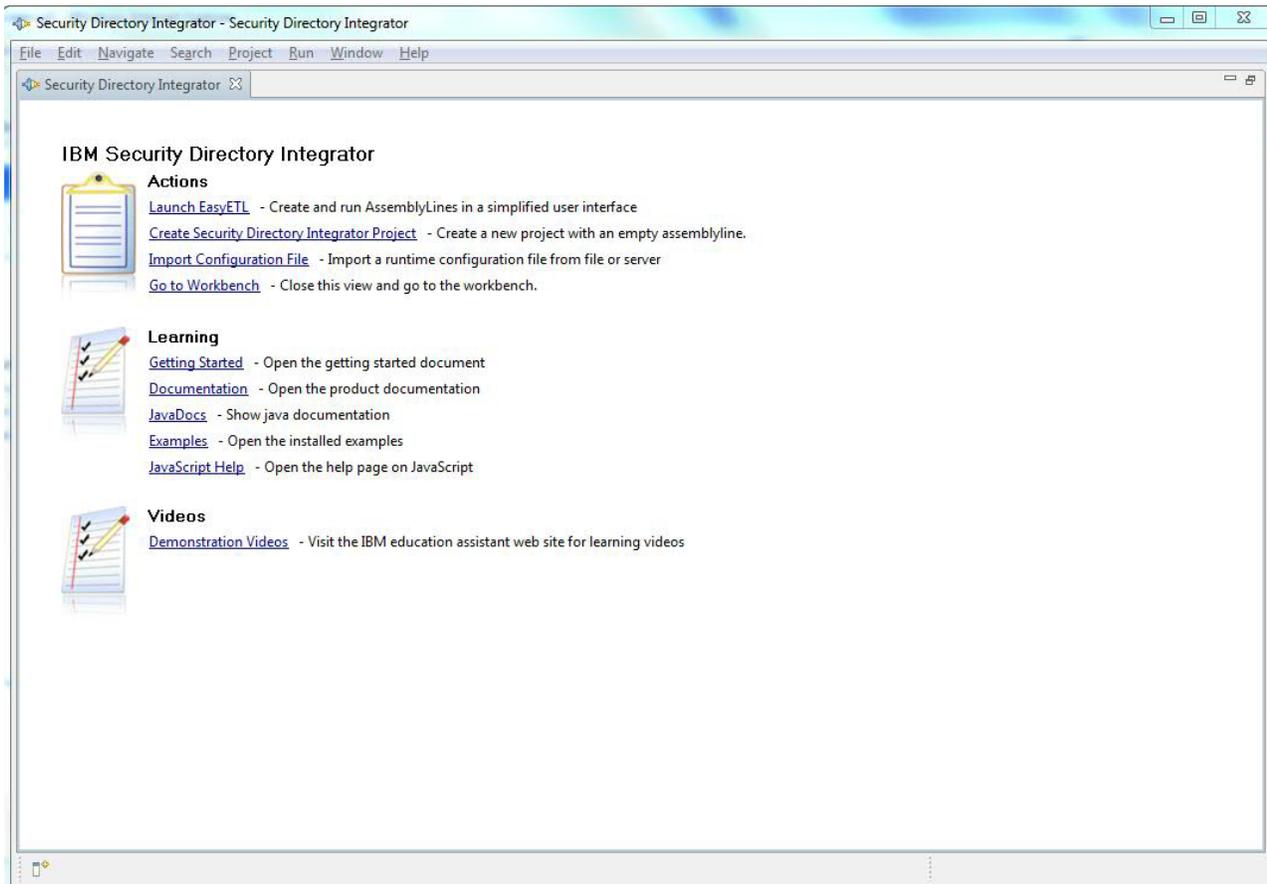


Figura 6. Pantalla de bienvenida del Editor de configuración

La pantalla de bienvenida ofrece una serie de enlaces de inicio rápido⁶.

Cuando cree, pruebe o modifique soluciones de integración con IBM Security Directory Integrator, trabajará dentro de un proyecto. Los proyectos son colecciones de líneas de ensamblaje y sus componentes y cada proyecto tiene su propia subcarpeta dentro del espacio de trabajo. Las líneas de ensamblaje y los componentes que constituyen un proyecto se almacenan como archivos individuales, que a su vez se ubican en subdirectorios de la carpeta de proyecto.

Seleccione el segundo enlace de la parte superior de esta página⁷ (*Crear proyecto de Security Directory Integrator*) para configurar el primer proyecto. A continuación, debe asignar un nombre al nuevo proyecto. Llámelo 'Tutorial' y pulse **Finalizar**.

6. Puede volver a esta pantalla en cualquier momento seleccionando **Ayuda > Bienvenida** en el menú principal.

7. El enlace situado en la parte superior, Launch EasyETL, abre un entorno de trabajo simplificado y aparece en el apéndice "Guía de EasyETL", en la página 89.

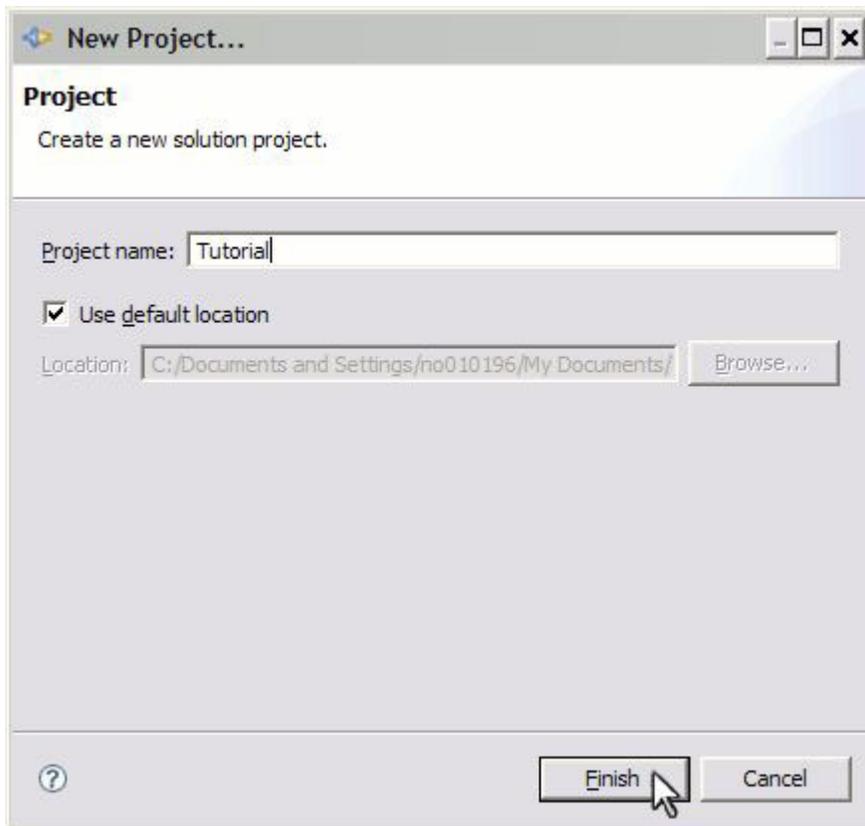


Figura 7. Dar un nombre al nuevo proyecto

A continuación verá el área de trabajo principal de desarrollo. Todos los paneles se pueden cambiar de tamaño, y puede decidir cómo organizar la pantalla. Lo que ve en la pantalla es la perspectiva predeterminada de IBM Security Directory Integrator⁸.

8. Una *perspectiva* es simplemente una organización determinada de los paneles del entorno de desarrollo. Si ha hecho cambios en el diseño y desea volver a la perspectiva predeterminada de IBM Security Directory Integrator, simplemente pulse **Ventana** en el menú superior y seleccione la opción **Restablecer perspectiva**.

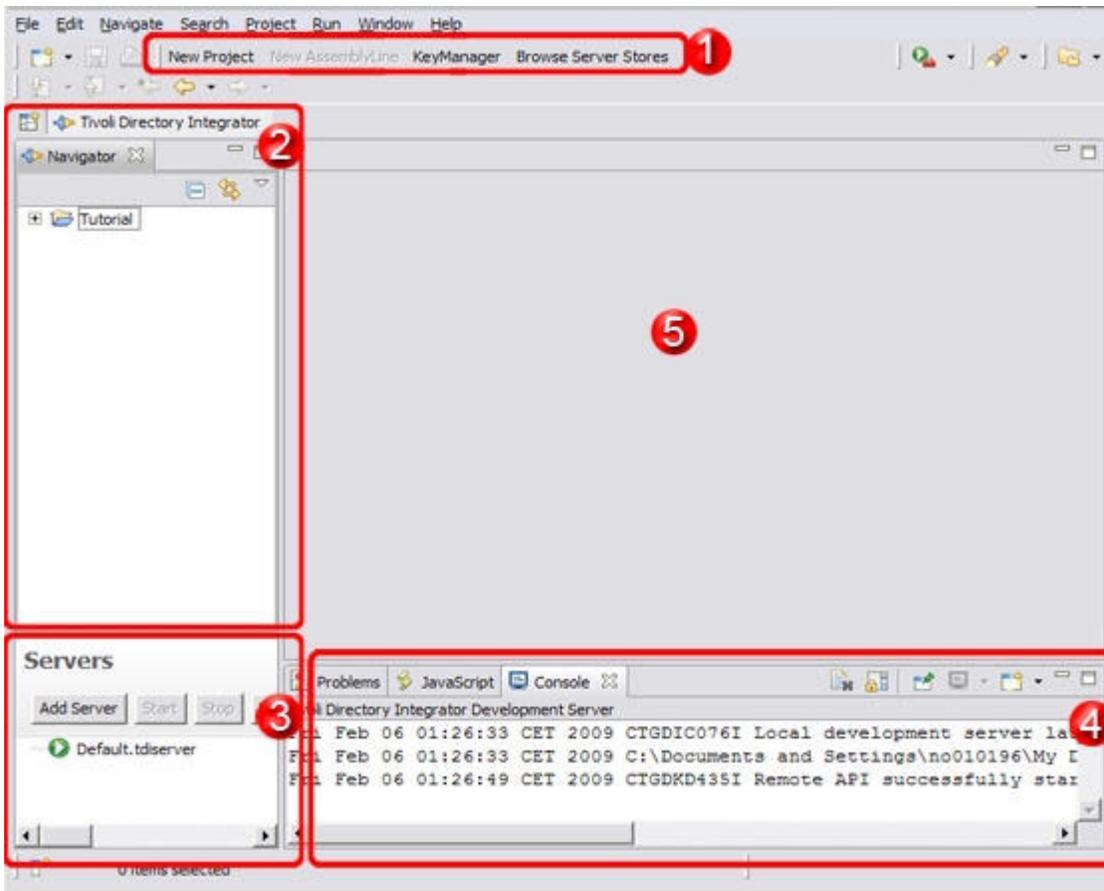


Figura 8. Pantalla principal del Editor de configuración

Nota: La única perspectiva que necesita utilizar es la perspectiva **Security Directory Integrator** o **Easy ETL**. Si el Editor de configuración no tiene la apariencia o los paneles que esperaba, siga los siguientes pasos:

- Seleccione **Ventana > Abrir perspectiva > Security Directory Integrator**. Esta opción selecciona la perspectiva Security Directory Integrator.
- Si no le es útil, seleccione **Ventana > Restablecer perspectiva....** Este mandato garantiza que todos los paneles predeterminados están colocados correctamente.

Esta es la pantalla principal donde pasará la mayor parte del tiempo cuando trabaje con IBM Security Directory Integrator. Sin entrar en detalles sobre todos los elementos de navegación que se pueden encontrar en esta pantalla⁹, comentaremos brevemente las distintas áreas numeradas que aparecen marcadas en la imagen:

1. En mitad de la fila de botones principal hay un conjunto de atajos para crear nuevos proyectos y, si hay un proyecto seleccionado en el navegador, para crear nuevas líneas de ensamblaje dentro de ese proyecto. También hay un botón para iniciar la herramienta KeyManager para trabajar con claves de certificados y almacenes de confianza; así como el botón **Examinar los almacenes de servidor** para recuperar los diversos valores de las propiedades del servidor de IBM Security Directory Integrator asociado a este proyecto.

9. Como ocurre en la mayoría de aplicaciones basadas en Eclipse, hay diversas formas de realizar la misma operación. La publicación *Configuración de Directory Integrator* describe las diversas opciones y paneles disponibles.

2. Éste es el panel *Navegador* y proporciona una vista de árbol de los activos de desarrollo. Su nuevo proyecto 'Tutorial' deberá aparecer aquí.
3. El panel *Servidores* muestra el estado de todos los servidores configurados. El icono en forma de flecha situado junto 'Default.tdiserver' indica que este servidor se ha iniciado automáticamente. Este panel también incluye botones para definir nuevos servidores, iniciar y detener servidores, así como para renovar la lista y ver el archivo de registro de un servidor¹⁰.

Observe que, cuando inicia una línea de ensamblaje, en este panel se muestran tanto la Instancia de configuración¹¹ como la LE.

4. En esta área puede ver un conjunto de pestañas, la que está seleccionada actualmente muestra la salida de la consola procedente del servidor. Los mensajes que se muestran en la imagen indican que el servidor está en ejecución y que su API se ha inicializado y está preparada para ser utilizada.
5. El área gris de la imagen es donde se abrirán los paneles del *editor* a medida que vaya creando líneas de ensamblaje y componentes. Cada tipo de recurso (Conector, Analizador, Línea de ensamblaje, etcétera) tiene su editor especialmente diseñado.

Cómo crear su primera línea de ensamblaje

Volviendo al caso de ejemplo mencionado en la introducción, en este ejercicio deberá crear una LE que migre información de D1 a D3, ignorando de momento la adición de datos procedentes de D2.



Figura 9. Diagrama simplificado del caso de ejemplo con sólo dos orígenes de datos

La carpeta 'Tutorials' (que debe haber copiado desde *directorio de instalación de TDI/examples* a su directorio de soluciones) contiene un archivo denominado `People.csv`:

```

First;Last;Title
Bill;Sanderman;Chief Scientist
Mick;Kamerun;CEO
Jill;Vox;CTO
Roger
Gregory;Highpeak;VP Product Development
Ernie;Hazzle;Chief Evangelist
Peter;Belamy;Business Support Manager

```

10. Si por algún motivo su servidor no se ha iniciado correctamente, abra 'Servidores de TDI' y efectúe una doble pulsación sobre 'Default.tdiserver'. Se abrirá el documento de servidor asociado. Asegúrese de que los valores de Directorio de instalación y de soluciones son correctos y pulse la opción **Crear directorio de soluciones** en la parte superior de este panel. Si con esto no se soluciona el problema, póngase en contacto con el servicio de soporte.

11. Cuando el servidor de IBM Security Directory Integrator carga una configuración, crea una *Instancia de configuración* que encapsula las líneas de ensamblaje de ese proyecto y permite que se ejecuten en su propio entorno contenido. Esto significa que puede cargar diversas veces la misma configuración en el mismo servidor, y se crearán Instancias de configuración distintas, cada una de las cuales contendrá el mismo conjunto de LE sin que interfieran unas con otras.

Como puede ver en el texto anterior, el archivo está en formato CSV (*character separated value*). Este archivo representa los datos de entrada del origen de datos D1. La LE que va a crear extraerá estos datos y los transferirá a un documento XML, que será el destino de salida D3.

Pulse en **Nueva línea de ensamblaje** en la barra de herramientas superior y escriba el nombre de la nueva línea de ensamblaje: 'CSV2XML'.

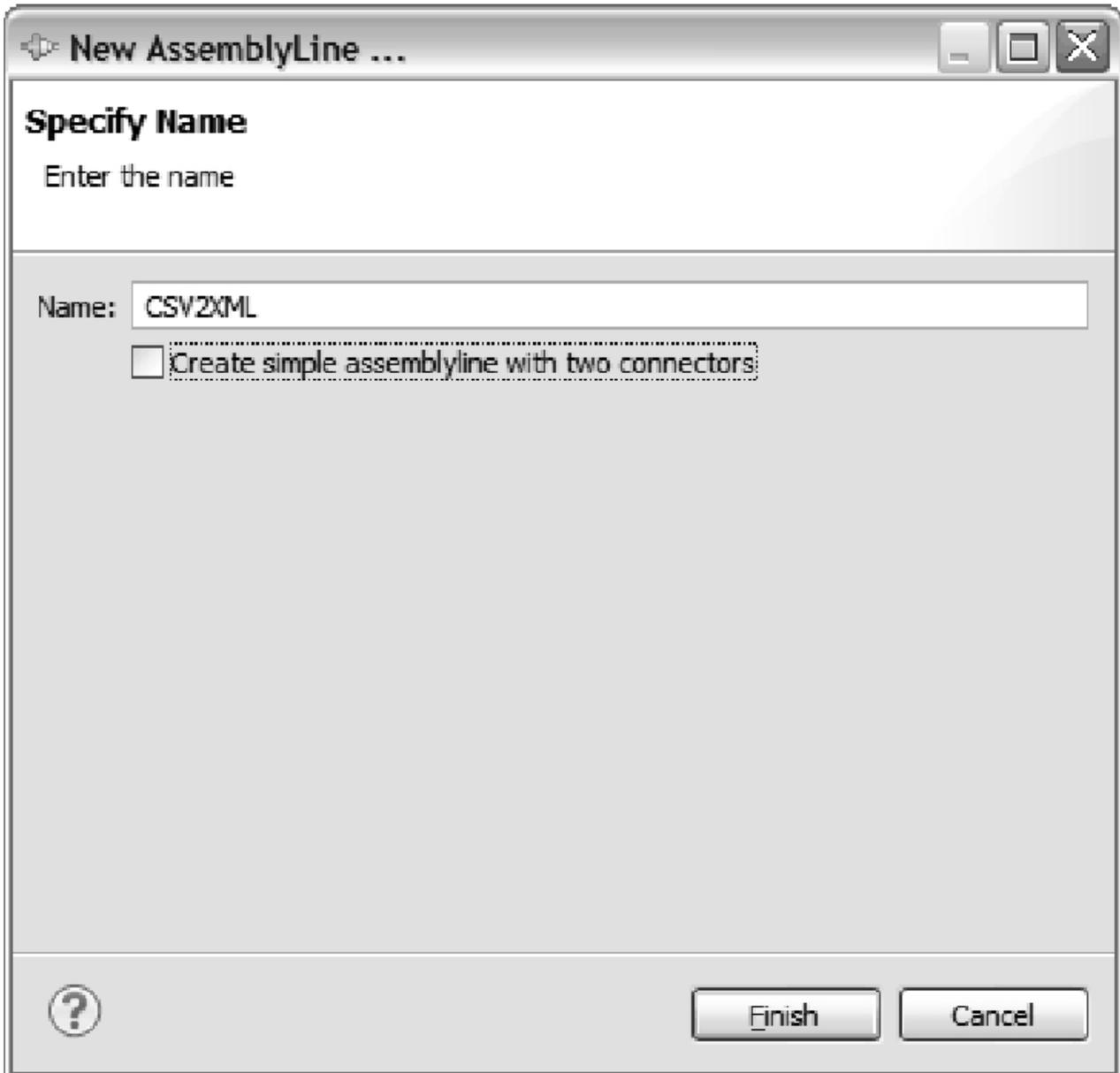


Figura 10. Recuadro de diálogo Nueva línea de ensamblaje

Ahora pulse el botón **Finalizar** para abrir la LE en una pestaña de editor de líneas de ensamblaje.

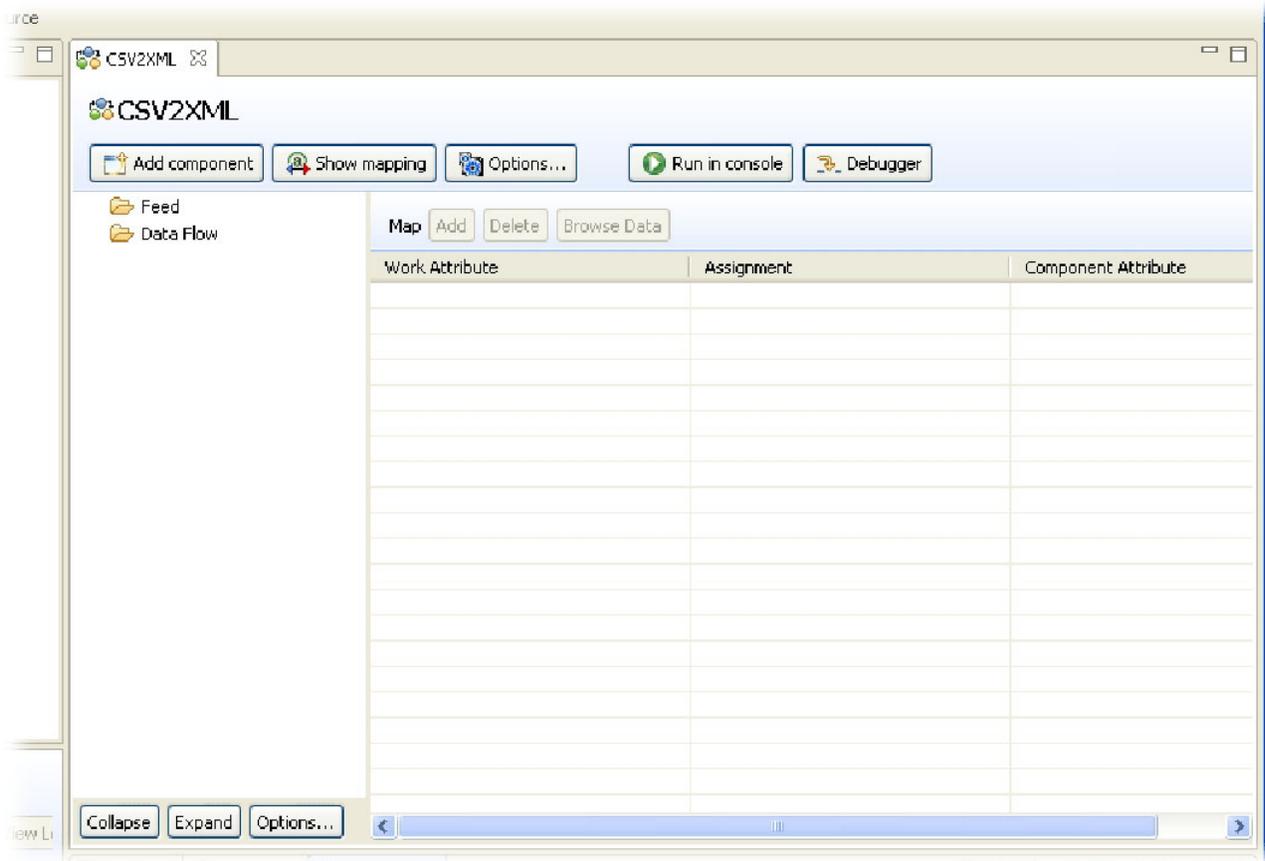


Figura 11. Editor de una línea de ensamblaje vacía

La parte izquierda de este editor de LE contiene la lista de componentes que configuran esta línea de ensamblaje y ahora mismo está vacía, aparte de los dos nombres de secciones: *Canal de información* y *Flujo de datos*. El área de la derecha muestra todos los atributos que se correlacionan como entrada y salida de la LE.

Para comprender estas secciones de la línea de ensamblaje, reflexione brevemente sobre lo que queremos que haga esta LE: *Para cada línea del archivo CSV, debe crear un nuevo nodo en el documento XML*. El kernel de IBM Security Directory Integrator proporciona este comportamiento en bucle automáticamente, transportando componentes desde la sección *Flujo de datos* de la línea de ensamblaje, mientras haya datos de entrada procedentes de los conectores, hasta la sección *Feed*¹².

Utilizaremos esta funcionalidad añadiendo un conector a la sección Canal de información para que lea nuestro archivo de entrada CSV. Para hacerlo, pulse con el botón derecho del ratón sobre la sección *Canal de información* y seleccione **Añadir Componente...**

12. Observe que solo un *conector de alimentación* entrega datos a la línea de ensamblaje cada vez. Si coloca más de un Conector Iterador, el primero se vaciará en primer lugar antes de que el siguiente empiece a leer de su origen.

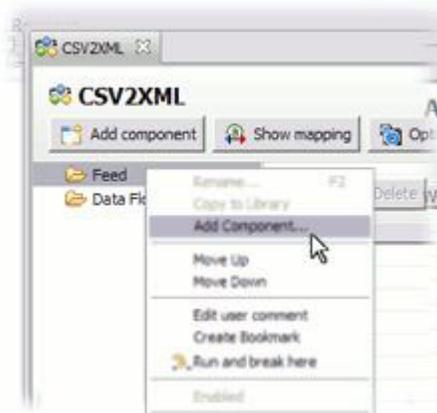


Figura 12. Añadir un nuevo componente

Se abrirá el asistente **Elegir componente**.

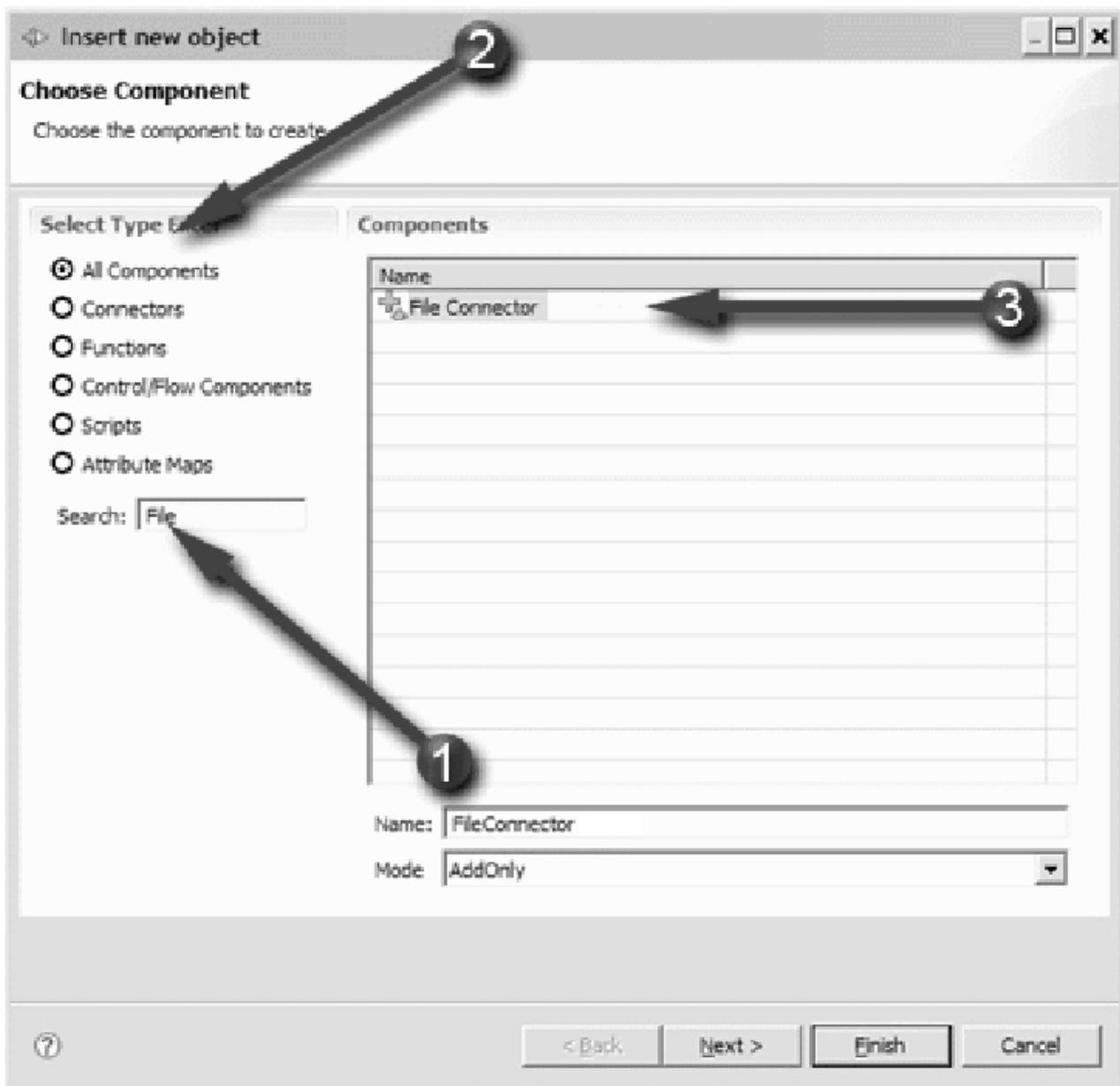


Figura 13. Elegir componente

Este diálogo le ofrece un par de opciones para encontrar y seleccionar el componente deseado:

1. Empiece a escribir cualquier parte del nombre del componente en el campo de texto y la lista de selección de la derecha se filtrará consecuentemente. Para este ejemplo, escriba "archivo".
2. Opcionalmente, puede limitar la lista de selección para incluir un único tipo de componente – Conectores, Analizadores, Scripts, etcétera.
3. Localice y seleccione el componente que desea de esta lista. En nuestro ejemplo será el 'Conector del sistema de archivos'.

El nuevo conector recibe automáticamente el nombre predeterminado 'Conectordelsistemadearchivos'. Cámbielo a 'Read_CSV_File' para que tenga más

significado en el contexto de su solución¹³ y seleccione **Iterador** en el desplegable **Modalidad**.

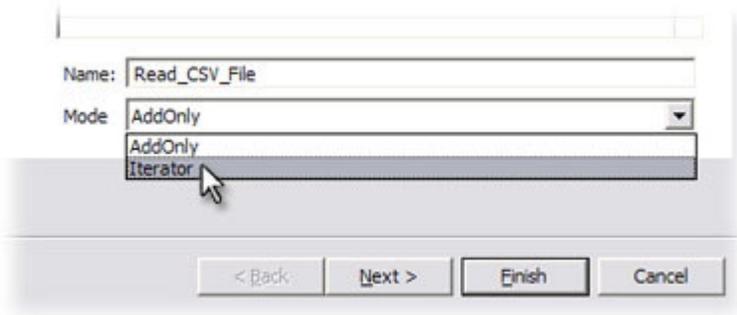


Figura 14. Renombrar el conector y cambiar la modalidad

El valor del campo Modalidad de un conector es el que indica a la lógica de ejecución incorporada en la LE el papel que ese componente juega en el flujo. La modalidad de iterador ofrece el comportamiento *for-each* que necesitamos para conducir los datos procedentes del archivo CSV, de una en una entrada, hacia los componentes que añadiremos en la sección *Flujo de datos*.

A continuación, pulse el botón **Siguiente** para pasar al panel de configuración del conector seleccionado.

13. Aunque puede poner cualquier nombre que desee a un conector, se recomienda que utilice nombres similares a los que se utilizan para las variables de scripts: un nombre que empiece por una letra, seguida por un número indeterminado de letras, dígitos y caracteres de guión bajo. El motivo es que todos los componentes de una LE se registran automáticamente como variables de scripts, de esta forma, podrá, si lo desea, reconfigurarlos y manejarlos después más fácilmente desde el código de script.

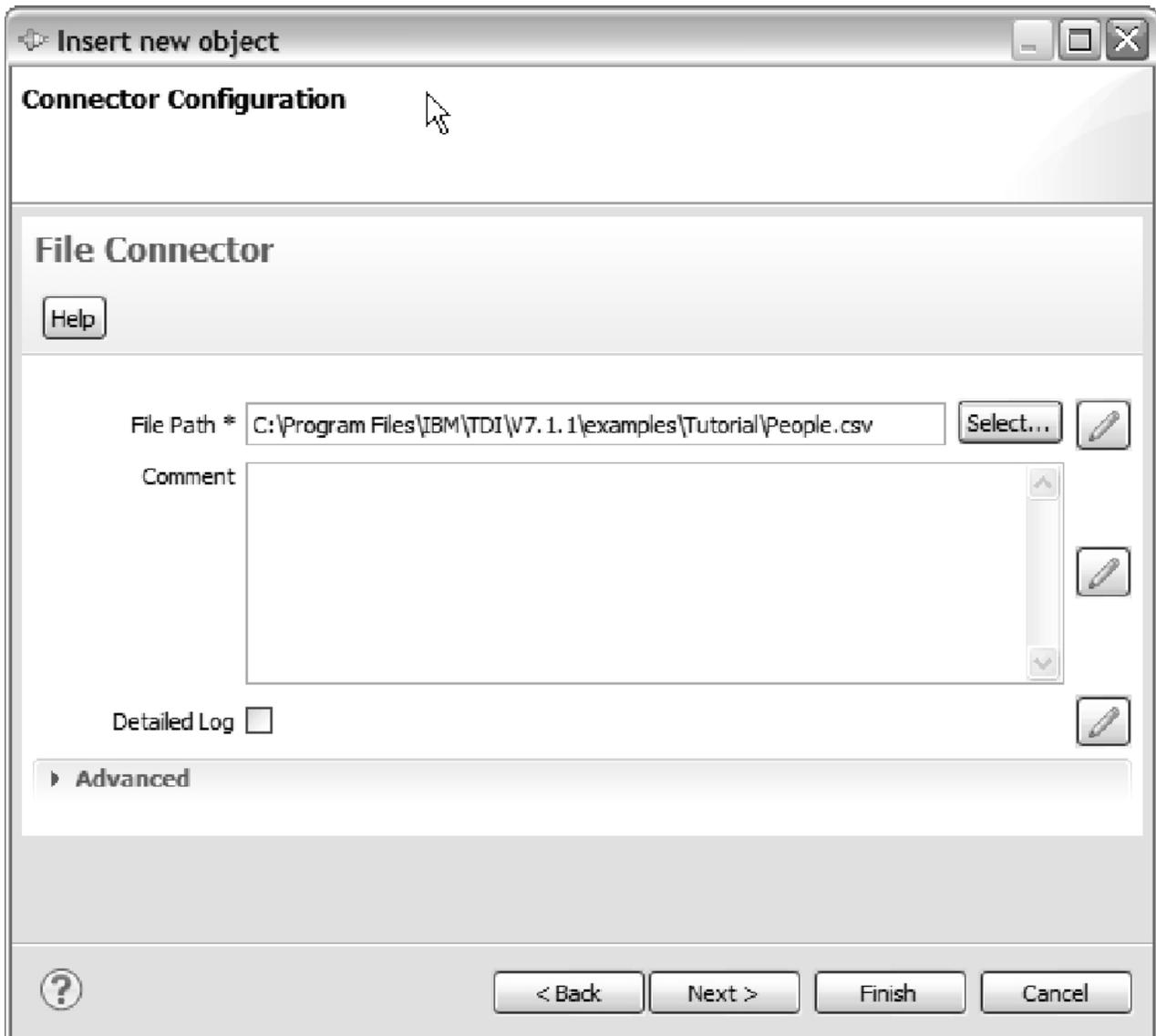


Figura 15. Panel de configuración del Conector del sistema de archivos

Cada componente proporciona su propio conjunto de parámetros de configuración. Los que se muestran ahora en pantalla son para el conector sistema de archivos y solo tiene un parámetro obligatorio: **Vía de acceso del archivo**. Escriba la vía de acceso al archivo `People.csv`, ya sea la vía de acceso completa o bien la vía de acceso relativa desde el Directorio de soluciones como se muestra en la imagen de arriba¹⁴ – o pulse el botón **Seleccionar** para abrir un explorador de archivos con el que buscar el archivo.

Dado que un archivo de texto con formato es una secuencia de bytes y no un origen de datos estructurado como una base de datos o un directorio, debe indicar un analizador que interprete el formato de la secuencia a medida que se va leyendo. IBM Security Directory Integrator proporciona una característica muy completa y versátil para Examinar datos, de forma que se puede probar de forma interactiva la selección y configuración del par conector/analizador.

14. Esta técnica facilita el poder mover o compartir una solución, ya que únicamente tendrá que especificar el Directorio de soluciones que desea y todas las vías de acceso relativas funcionarán sin necesidad de modificarlas.

Veremos esto más adelante, de momento debe completar este asistente volviendo a pulsar **Siguiente** para pasar al panel **configuración de analizador**. Aquí deberá pulsar en **Analizador CSV** para seleccionarlo.

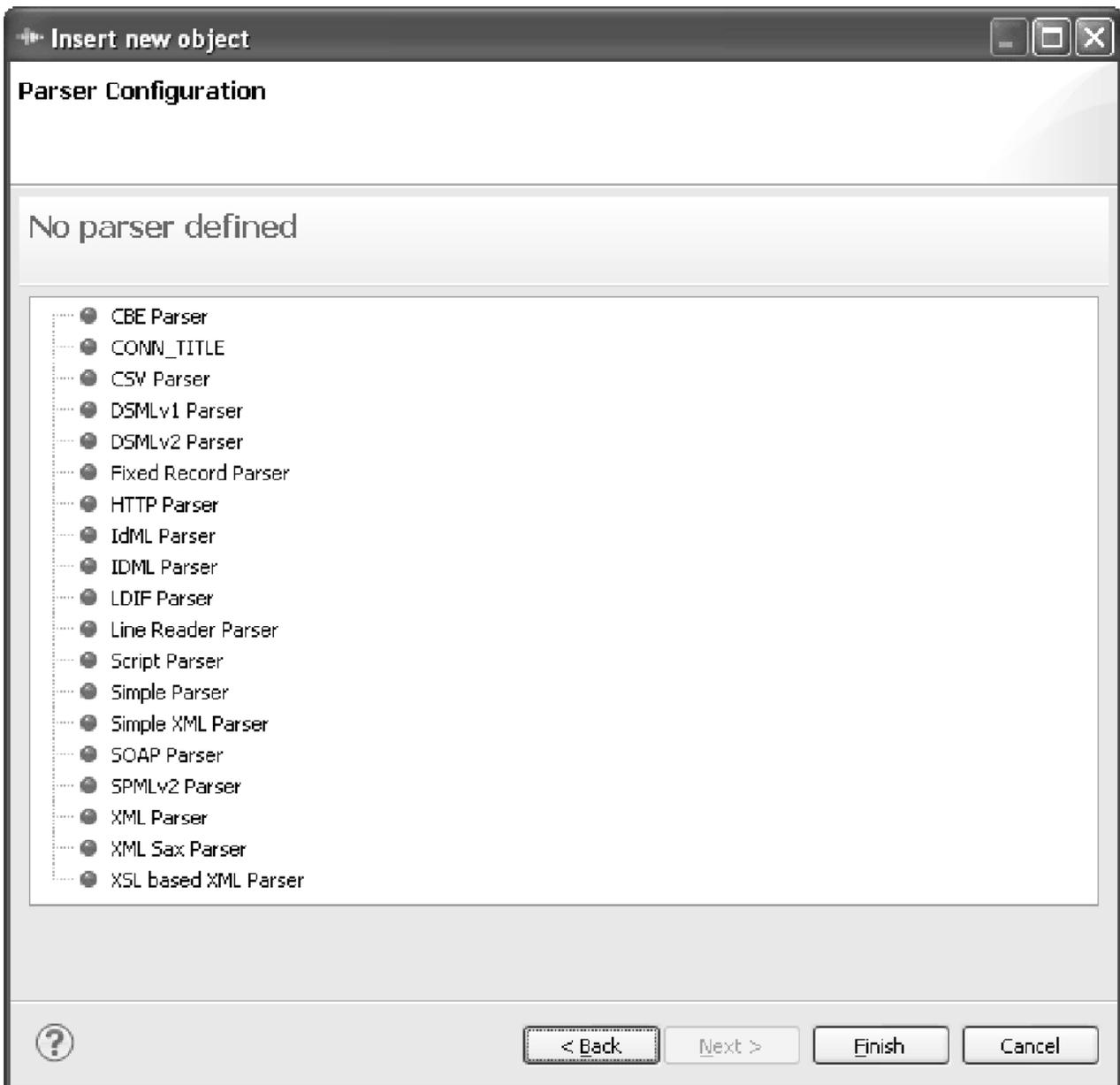


Figura 16. Selección del analizador al añadir un nuevo objeto

Una vez que haya seleccionado el analizador CSV, pulse **Finalizar** para cerrar el asistente. Volverá al panel Configuración de analizador. Dado que no necesita cambiar los valores predeterminados, vuelva a pulsar Finalizar para completar el asistente.

El siguiente paso es que el conector *descubra* el esquema del origen de entrada para correlacionar esos valores a la línea de ensamblaje. Es ahora cuando el Explorador

de datos resulta útil¹⁵. Inicie el Explorador de datos pulsando con el botón derecho del ratón sobre el nuevo conector iterador en el árbol de componentes de la línea de ensamblaje y seleccionando **Examinar datos** en el menú contextual.

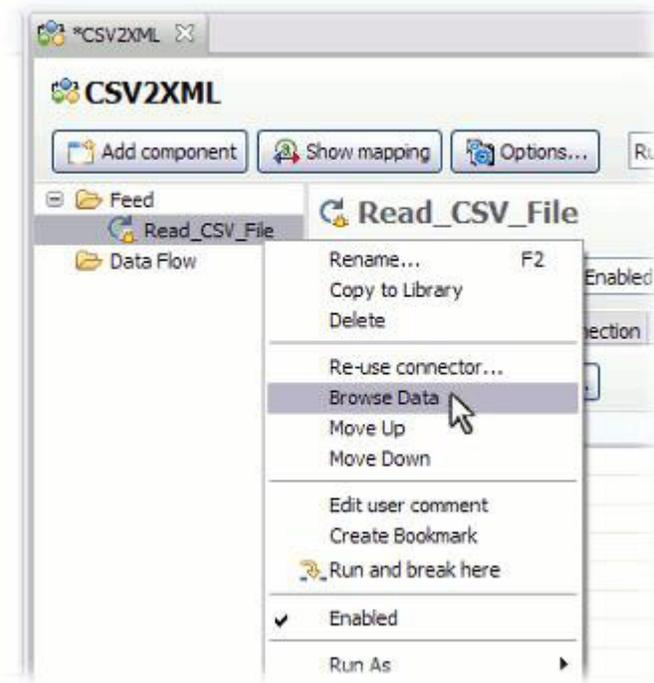


Figura 17. Selección del menú contextual Examinar datos

Se abrirá el Explorador de datos en una nueva pestaña del editor.

15. Puesto que sabe que el archivo está en formato CSV, la forma más rápida será pulsar los botones **Conectar** y **Siguiente** en el área Esquema del conector iterador. A continuación puede arrastrar los Atributos descubiertos en la Correlación de entrada como desee. El Explorador de datos resulta útil cuando no está seguro del formato. Pero aún así debería probarlo :)

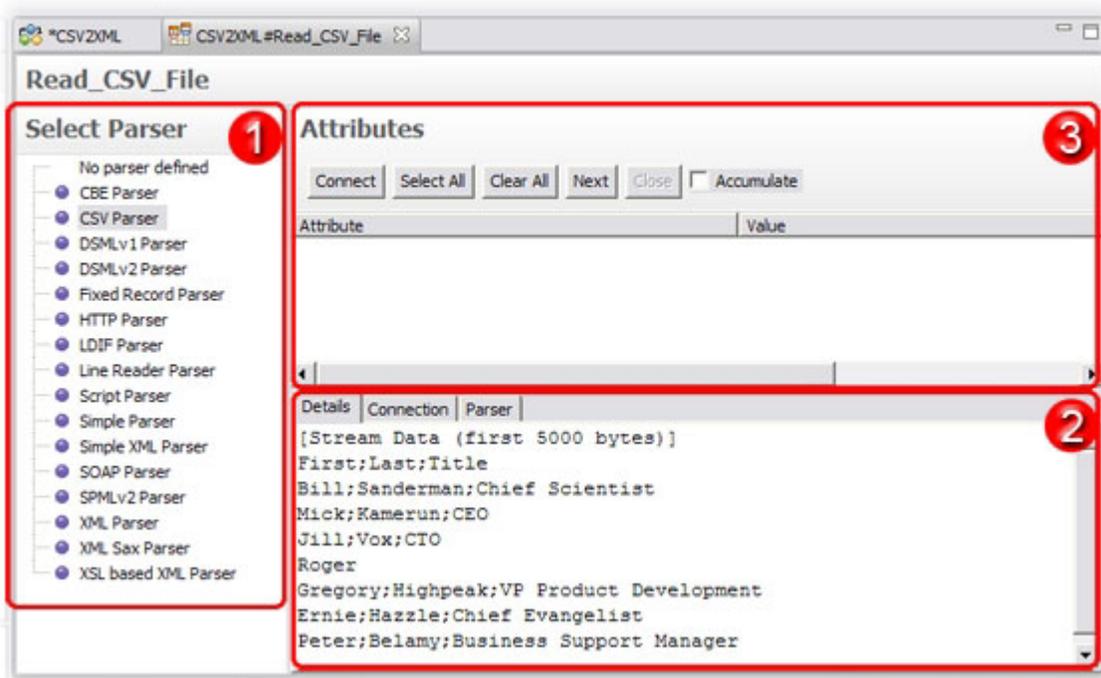


Figura 18. El Explorador de datos

El área 1 de la imagen es para elegir - y cambiar - el Analizador seleccionado. El área 2 proporciona una pestaña Detalles que le muestra la secuencia de bytes que se analizará. También dispone de una pestaña para cambiar los parámetros de conexión del conector y de otra pestaña para configurar el analizador seleccionado.

Por último, el área 3 es para conectar con el origen de datos y descubrir los atributos disponibles. Hágalo pulsando primero el botón **Conectar** y después el botón **Siguiente**.

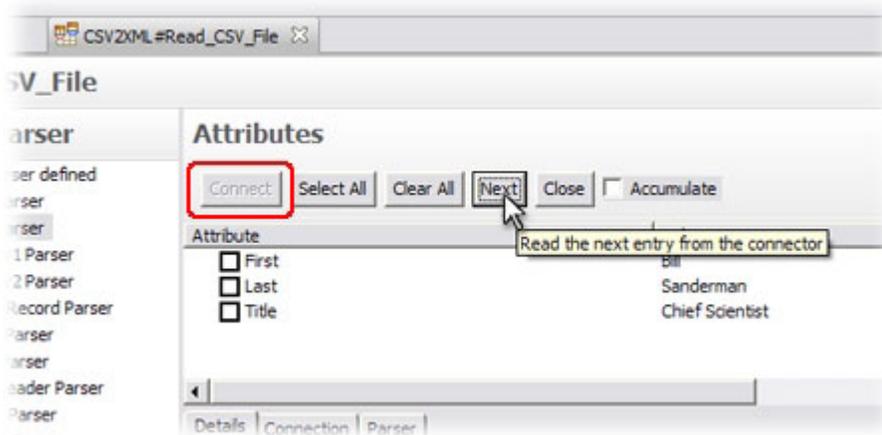


Figura 19. Descubrir el esquema de forma interactiva examinando los datos directamente

De esta forma, ha descubierto el *esquema* de este archivo. Seleccione los atributos que desea correlacionar, que en este caso son todos, seleccionando el recuadro de selección junto a cada uno de ellos, o utilizando el botón **Seleccionar todo**.

Utilice el atajo Ctrl-W para cerrar la pestaña Explorador de datos o pulse en el símbolo 'Cerrar' (X) en la esquina derecha de la pestaña y volver al editor de la línea de ensamblaje, donde la LE que tiene actualmente debería ser como la que se muestra en la imagen siguiente¹⁶.

Figura 20. LE con el conector iterador

La información detallada del componente seleccionado se muestran a la derecha de la lista de componentes de la LE, incluyendo las tres reglas de correlación que acaba de configurar en la Correlación de entrada. Cada elemento de Correlación de atributos tiene una **Asignación**, que es un fragmento de código de script que se evalúa para establecer el valor (o valores) del Atributo de destino.

Antes de continuar, deténgase un momento a reflexionar sobre estas *Asignaciones*: Recordará que en la sección “Modelo de datos Entrada-Atributo-valor” en la página 5 se ha explicado que la línea de ensamblaje tiene una *Entrada de trabajo* disponible a nivel global que contiene todos los datos que se transportan a lo largo de la LE. Para hacer referencia a este objeto en el código de script se utiliza la variable script predefinida *work*. Además la interfaz de cada Conector tiene su propia *Entrada de conector* que se utiliza como memoria caché para lecturas y escrituras. A este objeto específico de cada componente se accede desde el código de script mediante la variable predefinida *conn*¹⁷. A modo de ilustración, piense en la primera regla de correlación. Esta regla un atributo en la entrada de trabajo denominado 'First'. Su valor se deriva de la siguiente asignación:

```
conn.First
```

16. Si por algún motivo el Conector aparece en la sección *Flujo de datos* simplemente arrástrelo a la sección *Canal de información*. Si la Modalidad no es Iterador, pulse con el botón derecho del ratón sobre el Conector, seleccione **Modalidad** y después seleccione **Iterador**.

17. La variable *conn* sólo está disponible durante períodos limitados, como se muestra en los Diagramas de Flujo de Enganche de IBM Security Directory Integrator. Fuera de este ámbito se puede acceder a ella realizando una consulta a un componente sobre su Entrada de conector.

Esta notación abreviada hace referencia al atributo denominado 'First' que se acaba de leer en la entrada conn y sus valores se utilizan para llenar el atributo de la nueva entrada de trabajo. El script de asignación comparable sería:

```
return conn.getAttribute("First");18
```

Volviendo al ejercicio, ahora tiene que añadir el conector de salida para crear el documento XML de destino (el origen de datos D3). Esta vez, pruebe a utilizar el botón **Añadir componente** que encontrará en la parte superior del panel de Componentes de la línea de ensamblaje.

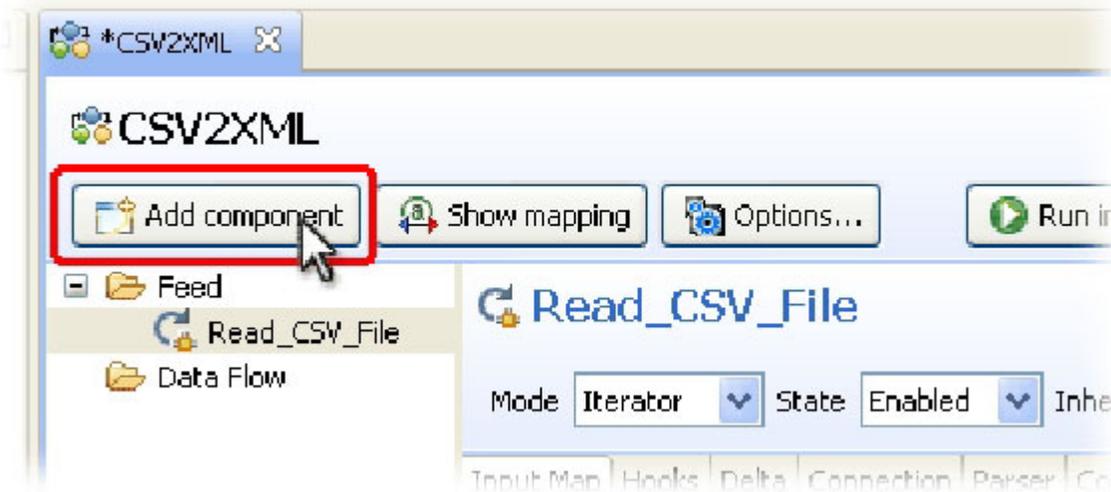


Figura 21. Botón Añadir componente

Vuelva a seleccionar el conector del sistema de archivos y renómbrelo como 'Write_XML_File'. Deje el valor Modalidad a **Sólo adición** y pulse **Siguiente**.

En el panel Configuración de conector, defina el parámetro **Vía de acceso del archivo** de forma que los datos se escriban en un archivo denominado `Output.xml` en la carpeta Tutorial. A continuación, elija 'Analizador XML' en el siguiente panel del asistente. Y después ya puede pulsar **Finalizar**, puesto que no necesita cambiar la configuración del Analizador XML. Observe que en el Conector de salida no puede realizar Descubrimiento del esquema porque no hay ningún archivo `Output.xml` desde donde descubrir datos.

18. Para los usuarios que estén familiarizados con la versión 6.x y anteriores, también se puede utilizar la sintaxis anterior a la versión 7.0:

```
ret.value = conn.getAttribute("First")
```

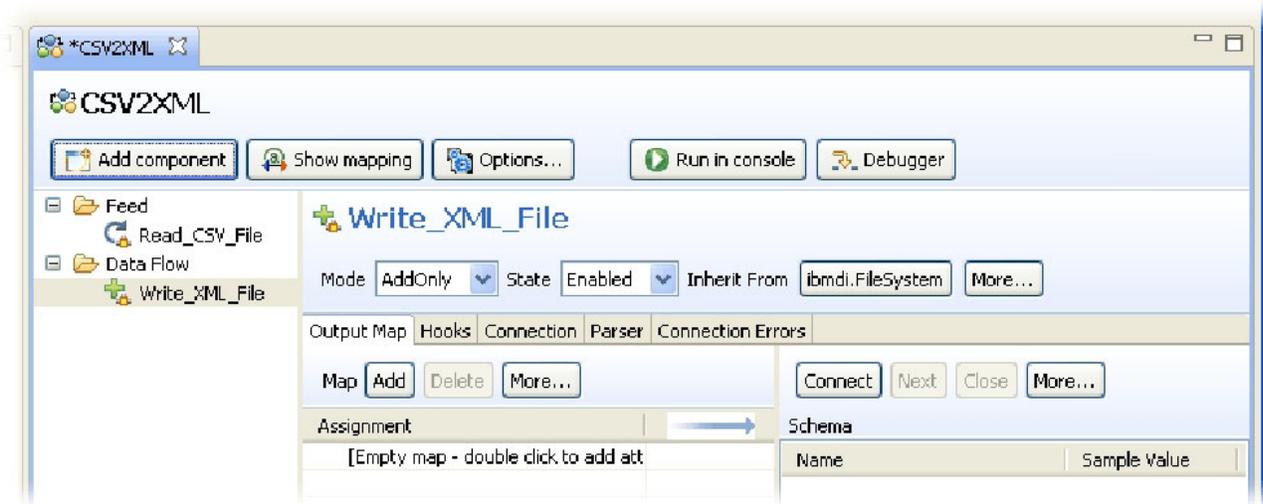


Figura 22. LE con los dos conectores

Quizás ya haya observado que cuando selecciona un componente, su información detallada se muestra en la parte derecha de la pantalla del editor. Cuando selecciona la carpeta 'Canal de información' o bien la carpeta 'Flujo de datos', se muestra una visión general de todas las **Correlaciones de atributos** para esta línea de ensamblaje. Esta visualización resulta muy útil para copiar los atributos de entrada a la correlación de salida de su último conector, así pues, abra esta visualización pulsando en 'Canal de información' o en 'Flujo de datos'.

Aquí puede ver una lista de atributos (tres en total) que el conector de modalidad iterador introduce en la LE. Seleccione estos Atributos de la Correlación de entrada¹⁹ y arrástrelos a la Correlación de salida del conector 'Write_XML_File', completando el flujo de datos.

19. Puede seleccionar varios atributos sueltos pulsando el botón del ratón mientras mantiene pulsada la tecla Control o seleccionar varios atributos seguidos pulsando el botón del ratón mientras mantiene pulsada la tecla Mayús.

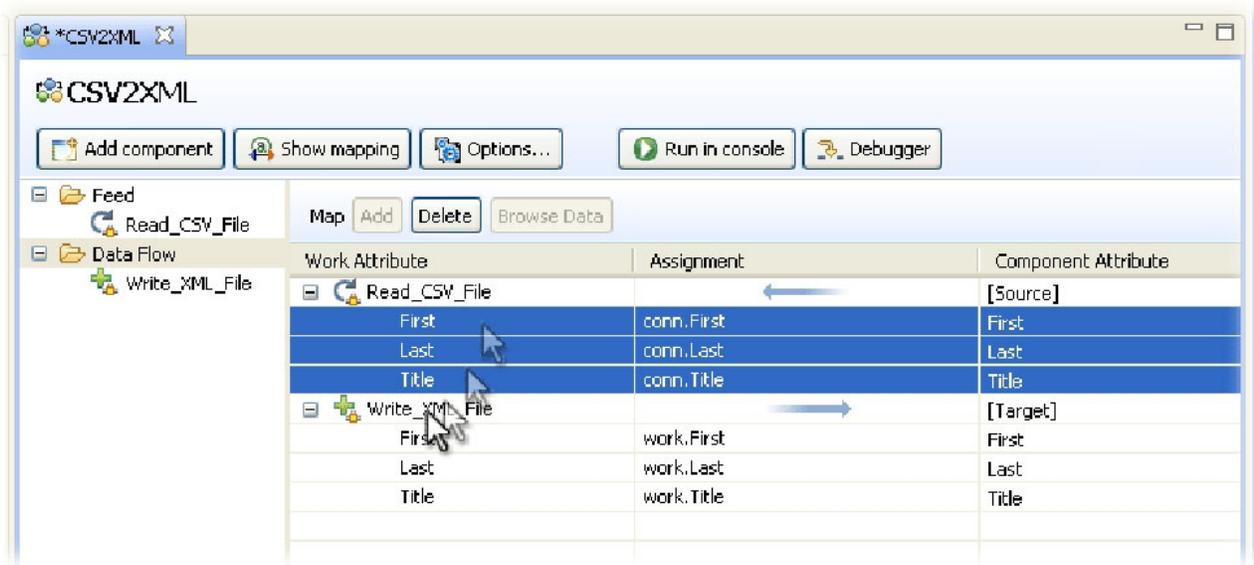


Figura 23. Arrastrar los atributos a la Correlación de salida

Observe que la asignación se convierte automáticamente de formato de entrada a formato de salida. Por ejemplo, el primer elemento de la Correlación de entrada del conector 'Read_CSV_File' creará un atributo en la entrada de trabajo denominado 'First' para guardar los valores que encuentre en conn.First (es decir, en el atributo denominado 'First' que se ha leído en la Entrada de conector). Cuando arrastra esta regla de correlación de entrada a una Correlación de salida, se cambia su asignación de forma que el valor ahora provenga de la entrada de trabajo y cree un atributo de destino en la memoria caché del conector (la Entrada de conector).

Si desea cambiar el origen en cualquier regla de correlación, debe editar la asignación. Para cambiar el nombre de los atributos a los que se correlaciona, pulse sobre un atributo con el botón derecho del ratón y renómbrela. Hágalo para las dos primeras reglas de la correlación de salida.

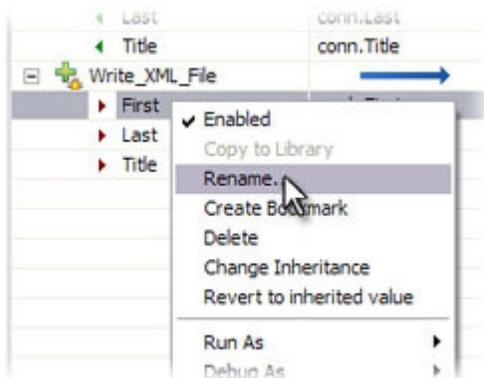


Figura 24. Renombrar una regla de correlación de atributos

Cambie 'First' a 'FirstName' y 'Last' a 'LastName'.

A continuación, añada un nuevo elemento a esta correlación de salida pulsando con el botón derecho del ratón en la correlación de salida 'Write_XML_File' y

eligiendo la opción **Añadir atributo**.

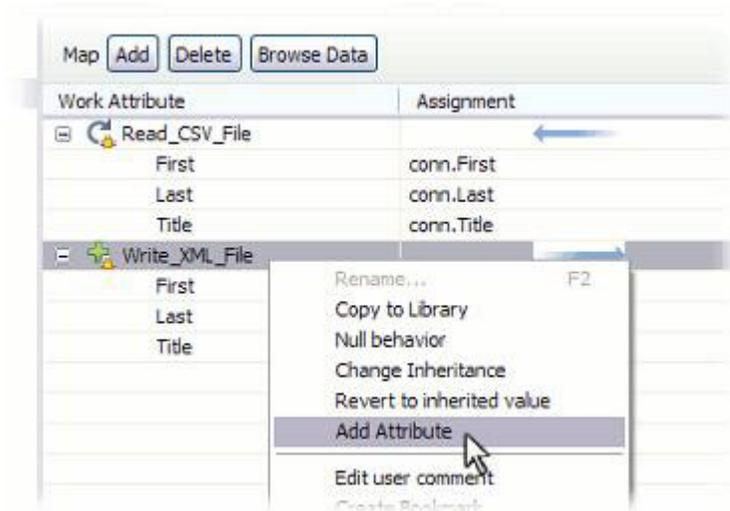


Figura 25. Añadir el atributo 'FullName' a la correlación de salida

Como nombre para el destino de esta nueva regla de correlación, escriba 'FullName', pulse **Aceptar** y efectúe una doble pulsación sobre este elemento para editar su asignación. Se abrirá el panel de editor de scripts que muestra un script de asignación predeterminado: `work.FullName`. Como no hay ningún atributo 'FullName' en la entrada de trabajo, esta correlación no podrá asignar ningún valor. En vez de utilizar el atributo 'FullName', deberá *calcular* este valor cambiando el script de forma que concatene los atributos First y Last, dejando un espacio en blanco entre ambos valores:

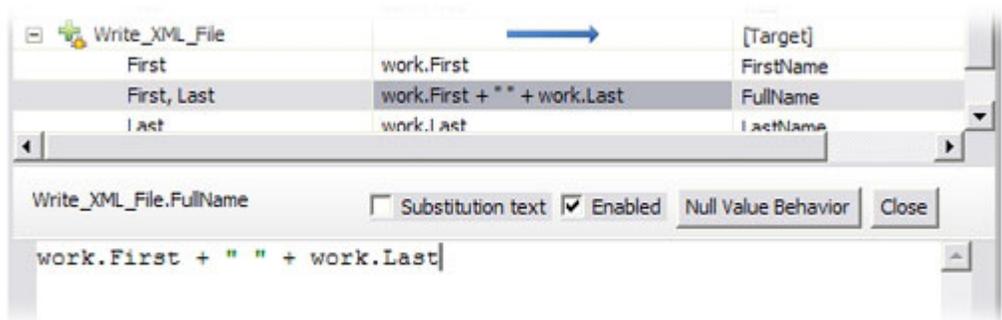


Figura 26. Editar la asignación

El script debe quedar como sigue:

```
work.First + " " + work.Last
```

Observe que no es necesario el punto y coma final para los scripts de asignación de correlación de atributos de una sola línea como éste²⁰. Pulse el botón **Cerrar** en la parte superior derecha del panel del editor de scripts cuando haya terminado.

20. Se admite también la sintaxis anterior a la versión 7.0, esto significa que los scripts de asignación de correlación pueden empezar también con "ret.value =".

Ejecutar la línea de ensamblaje

Ha llegado el momento de probar la LE.

Para hacerlo, pulse el botón **Ejecutar** que verá en la parte superior del editor de líneas de ensamblaje.



Figura 27. El botón Ejecutar

Verá que se abre una nueva pestaña con una *ventana de ejecución* que muestra la salida del registro de la línea de ensamblaje.

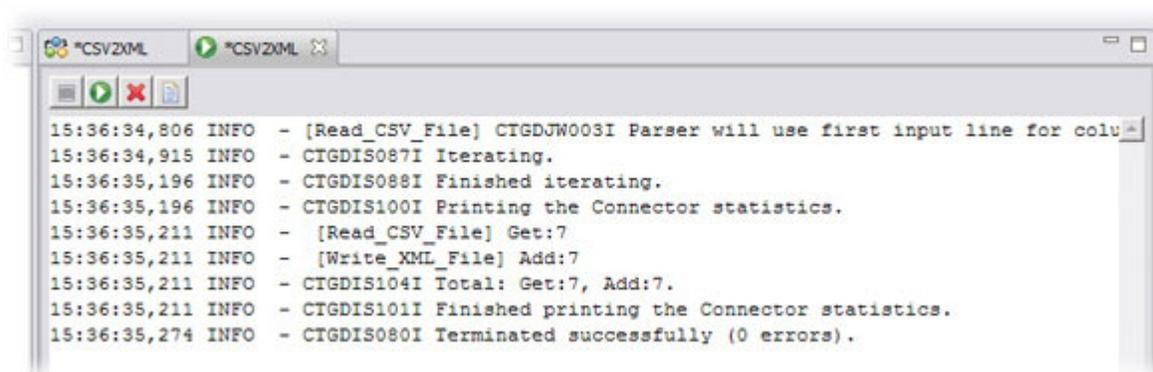


Figura 28. Salida del registro de la ejecución de la línea de ensamblaje

En realidad, el Editor de configuración ha cogido la línea de ensamblaje con todos sus componentes y ha exportado un *Config*, que es un documento XML que define el trabajo asignado a un servidor de IBM Security Directory Integrator. Después ha conducido este Config al Servidor y le ha indicado que ejecute la línea de ensamblaje, capturando el archivo de registro para mostrarlo en pantalla.

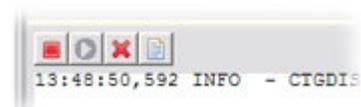


Figura 29. Barra de botones de la ventana Salida del registro

La ventana Ejecutar incluye una barra de botones con opciones para detener la LE, para reiniciarla una vez detenida y para borrar el contenido del registro. El botón que se encuentra más a la derecha abre la salida del registro actual en un editor externo.

La salida del registro de una línea de ensamblaje termina con unas estadísticas de todos los componentes participantes, que en su caso son sólo dos conectores. A partir de la información de la imagen, vemos que se han leído siete entradas del archivo CSV y se han escrito siete nodos en el documento XML.

Puede guardar el archivo de salida en el disco y abrirlo en una ventana del explorador para confirmar los resultados. También puede utilizar el Explorador de datos pulsando con el botón derecho del ratón en el conector de salida (Write_XML_File) y seleccionando **Examinar Datos**.

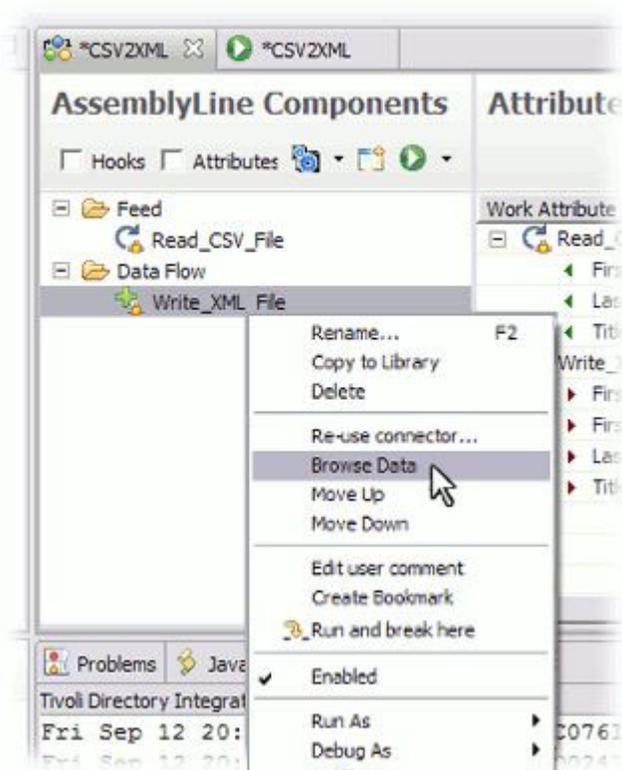


Figura 30. Examinar datos creados por un conector de salida

Se abre el Explorador de datos para el conector elegido, configurado y a punto para empezar. Pulse el botón **Conectar** y después pulse **Siguiente** para leer y mostrar los datos de salida.

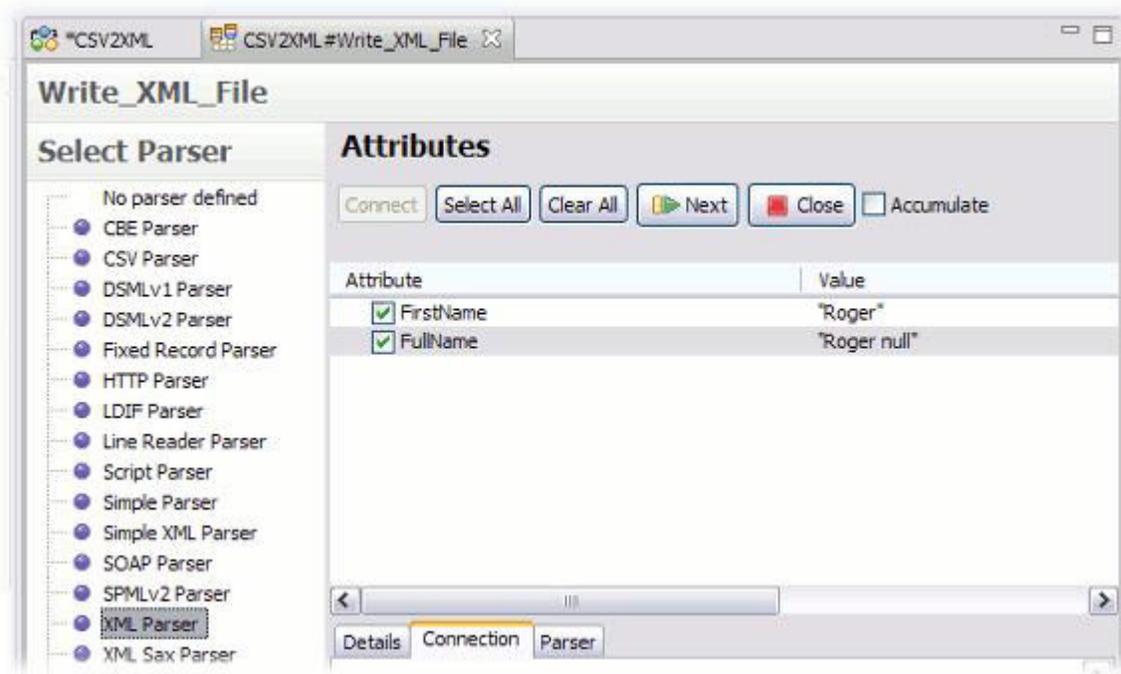


Figura 31. Examinar el XML resultante

El XML de salida debe ser una representación exacta de los valores de entrada, más la lógica de correlación; todo parece estar bien excepto la cuarta entrada (Roger), a la que le faltan los valores de 'LastName' y 'Title', y tiene como valor calculado para 'FullName': Roger null.

Si examina los datos de entrada más atentamente, verá que una de las líneas del archivo CSV está incompleta:

```

First;Last;Title
Bill;Sanderman;Chief Scientist
Mick;Kamerun;CEO
Jill;Vox;CTO
Roger
Gregory;Highpeak;VP Product Development
Ernie;Hazzle;Chief Evangelist
Peter;Belamy;Business Support Manager

```

El hecho de que falten datos o de que los datos no sean válidos es un fenómeno frecuente; su solución deberá estar preparada para filtrar o corregir este problema durante el proceso.

Comportamiento ante un valor nulo: Cuando faltan atributos o valores

Para definir el comportamiento cuando faltan valores, puede utilizar la característica incorporada de Comportamiento ante un valor nulo de las correlaciones de atributos, o puede detectarlo y manejarlo usted mismo.

Empezaremos configurando el Comportamiento ante un valor nulo. Para hacerlo, pulse con el botón derecho del ratón en el conector 'Read_CSV_File' del panel Correlaciones de atributos (no en la vista de árbol de componentes de la LE) y

seleccione **Comportamiento ante un valor nulo** en el menú contextual²¹.

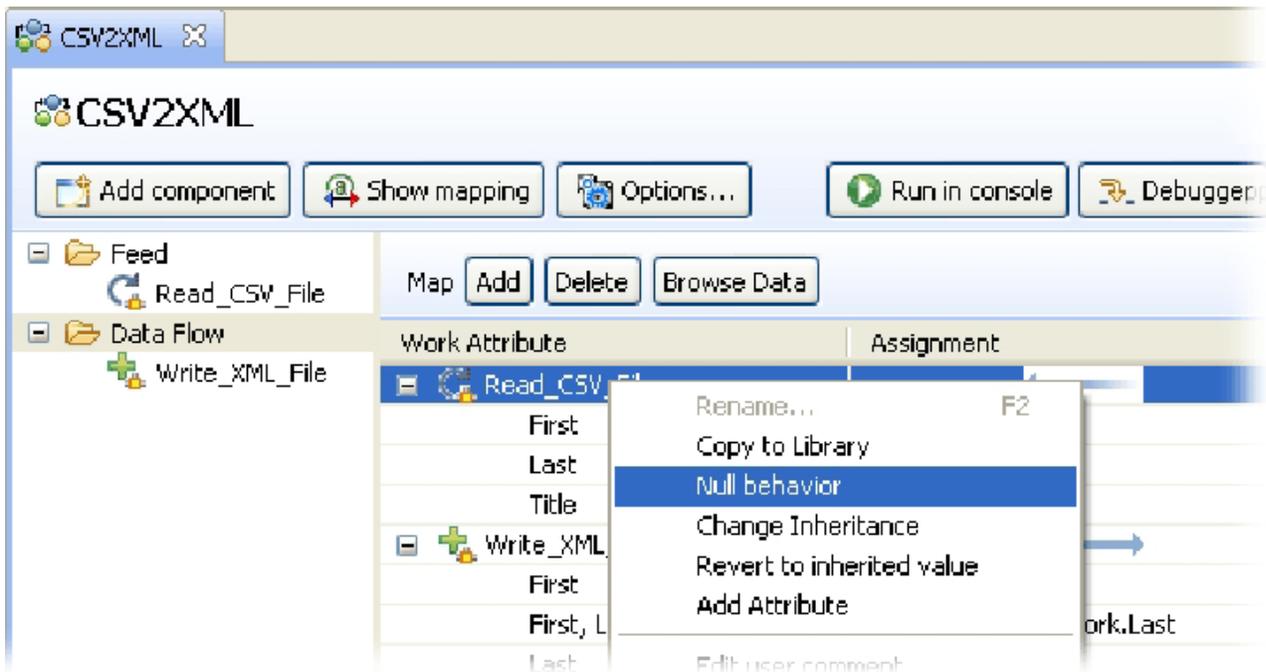


Figura 32. Botón Comportamiento ante un valor nulo para la configuración a nivel de la LE

Se abrirá el diálogo Comportamiento ante un valor nulo donde podrá definir qué se entiende por *nulo* – concepto que puede variar según el tipo de origen desde el cual se esté leyendo – y también cómo se debe manejar esta situación. De forma predeterminada, *nulo* significa que falta un atributo y el comportamiento predeterminado es eliminar este atributo de la operación de correlación. Con el resultado final de que no se encontrará ningún atributo con el nombre especificado en la entrada que se reciba.

Una vez en el diálogo Comportamiento ante un valor nulo, utilice los botones de selección de la derecha para definir *nulo* como un valor de serie vacía y los de la izquierda para especificar que se devuelva el valor predeterminado `"* missing *"` en ese caso.

21. También puede seleccionar el conector mismo en la lista de componentes de la LE, pulsar el botón **Más...** en la parte superior de la Correlación de entrada y elegir allí la opción 'Comportamiento ante un valor nulo'.

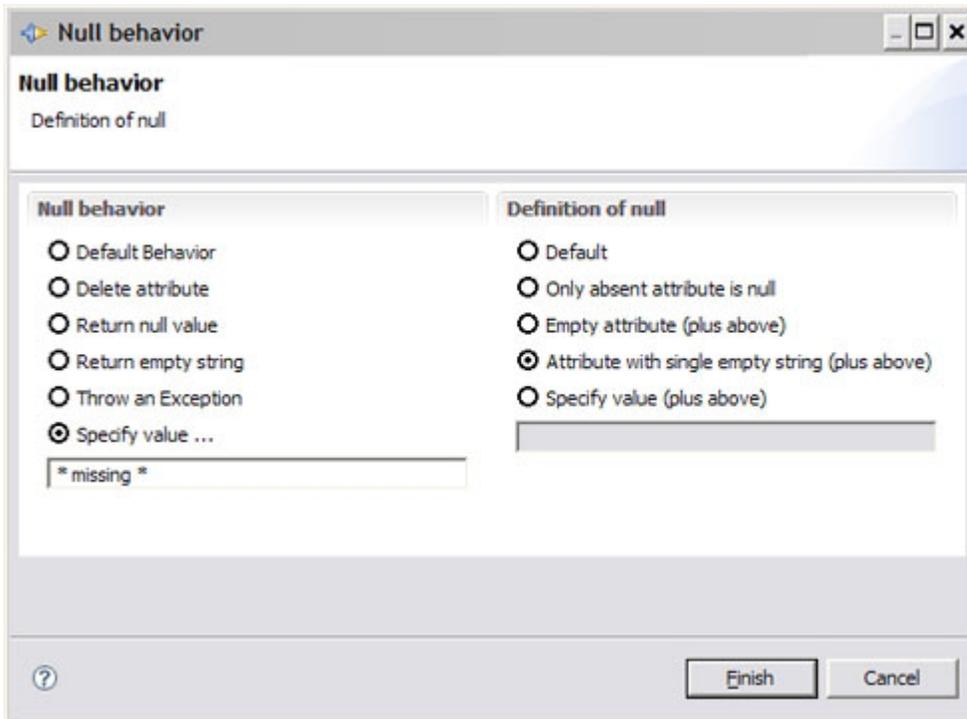


Figura 33. Diálogo de comportamiento del Comportamiento ante un valor nulo

Vuelva a ejecutar la línea de ensamblaje y renueve la ventana del explorador para mostrar el contenido de Output.xml. Ahora, la entrada de 'Roger' debería tener el valor *nulo* especial ("* missing *") para 'Title' y 'LastName'.

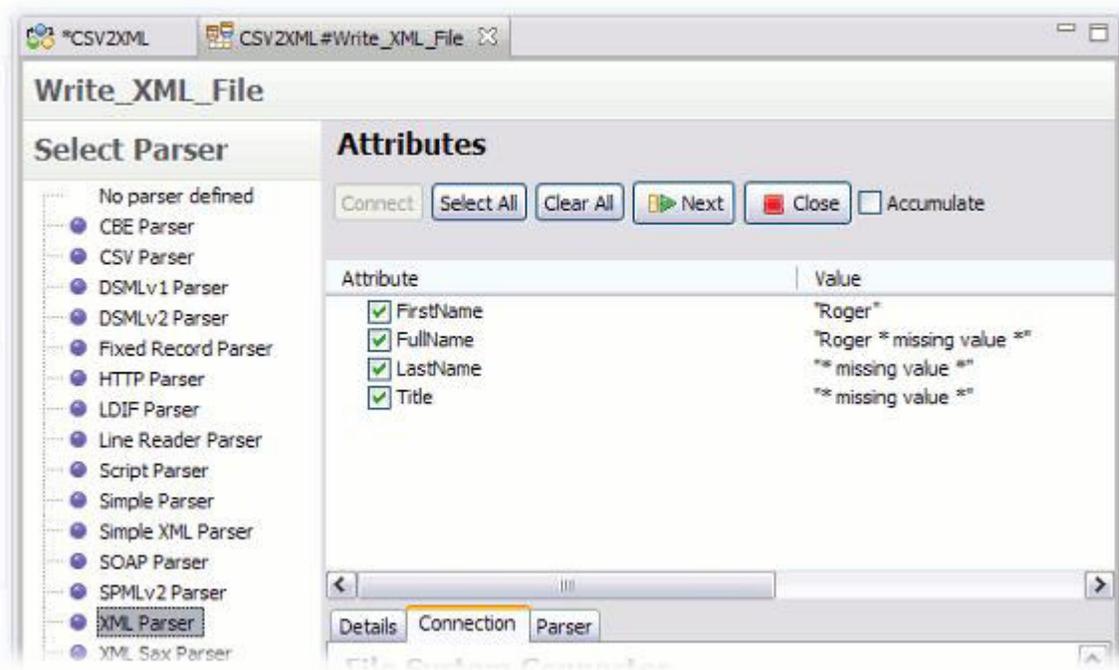


Figura 34. Resultado en la salida XML de los valores de Comportamiento ante un valor nulo

Esto ya está mejor – por lo menos es una elección consciente. Sin embargo, a veces una entrada está demasiado incompleta para continuar procesándola. En su caso de ejemplo, necesita por lo menos valores para 'FirstName' y 'LastName' para poder calcular 'FullName', así que ahora añadiremos lógica de filtrado a la línea de ensamblaje para asegurar que todas las entradas cumplen este requisito.

Para empezar, vuelva a pulsar el botón **Añadir componente** y después seleccione **Componentes de Control/Flujo** en los botones de selección de la izquierda. Seleccione 'IF' en la lista, ponga el nombre 'Incomplete data'²² y pulse **Finalizar**.

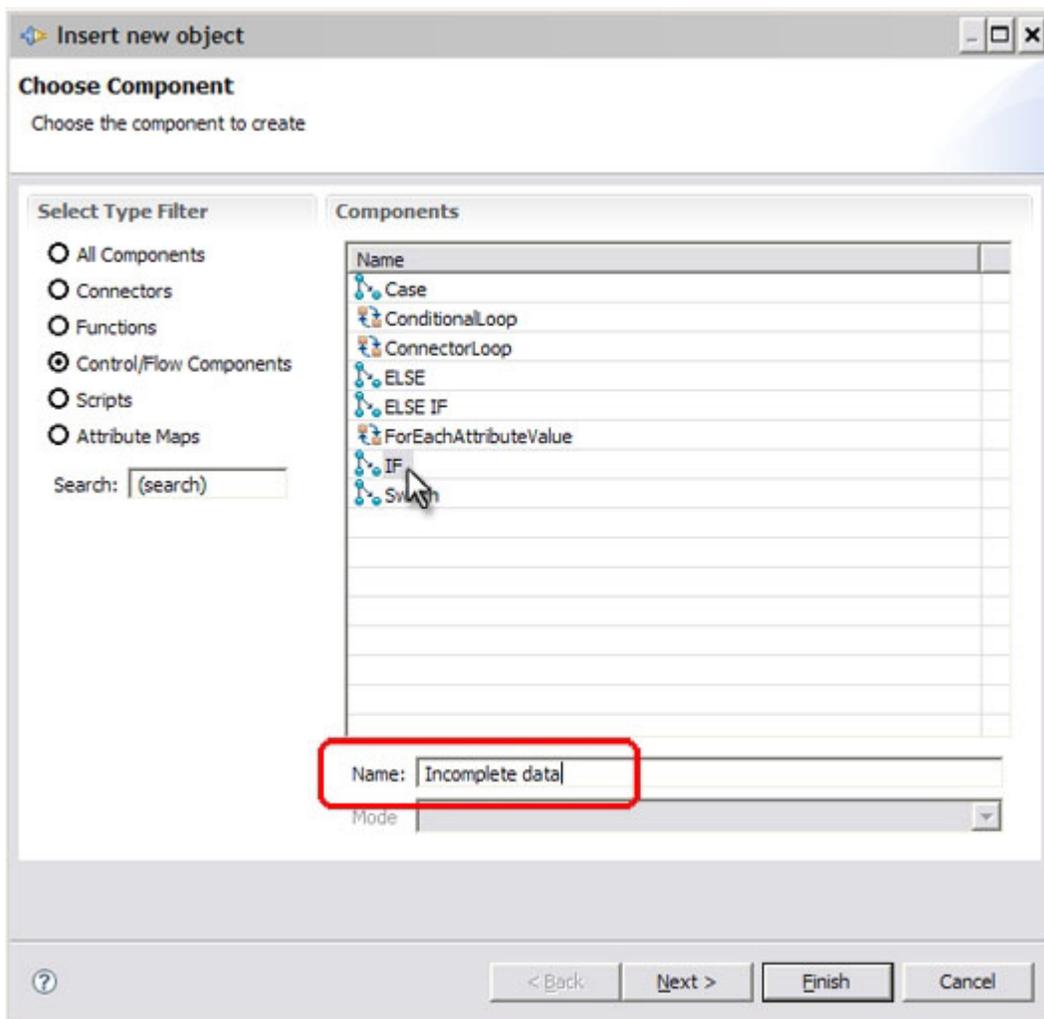


Figura 35. Seleccionar el componente de rama IF

A continuación, arrastre esta rama IF por encima del conector de salida. El área del editor de la rama IF le permite añadir condiciones.

22. Anteriormente se le ha indicado que es mejor que el nombre que ponga a los conectores sea como el de una variable de script. Esto también se aplica a los componentes de función. Sin embargo, no es tan importante para los componentes de una correlación de atributos, ramas, bucles y scripts, puesto que pocas veces se accede a éstos elementos desde un script. En esta guía se ha dado más prioridad a que la línea de ensamblaje sea fácil de leer.

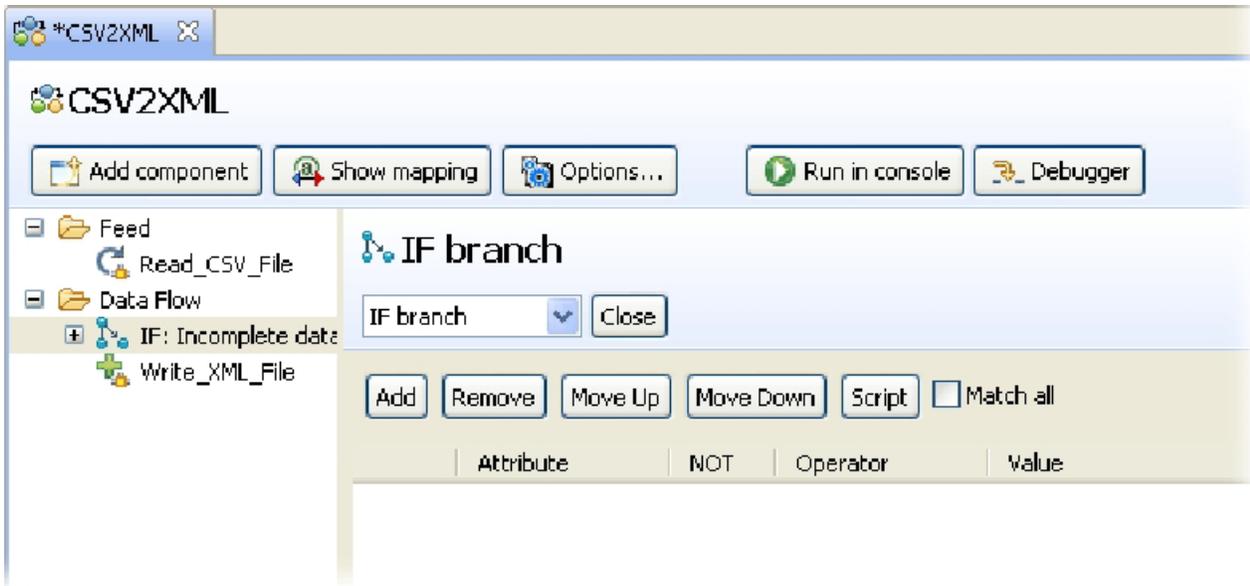


Figura 36. Editar condiciones para la rama IF

Aquí tiene la opción de añadir condiciones simples o escribir un fragmento de código script - o ambas cosas. Observe el recuadro de selección **Coincidir todo**, que decide si las condiciones (simples y en script) se evalúan con un *OR* implícito entre ellas (cuando el recuadro no está seleccionado) o con un *AND* (cuando sí lo está).

Añada una condición simple pulsando el botón **Añadir**. Después seleccione el atributo 'First' en el desplegable de la izquierda y después el operador 'tiene el(los) valor(es)'. Niegue esta condición conmutando el valor de la columna **NO**. No es necesario especificar nada en el campo de la derecha cuando se utiliza el operador 'tiene el(los) valor(es)'. A continuación, añada otra condición *no tiene valores* también para el atributo 'Last'. Finalmente, asegúrese de que el recuadro **Coincidir todo** no está seleccionado – lo que significa *Coincidir cualquiera* – de forma que, si faltan valores para cualquiera de los atributos, se desencadenará esta rama.

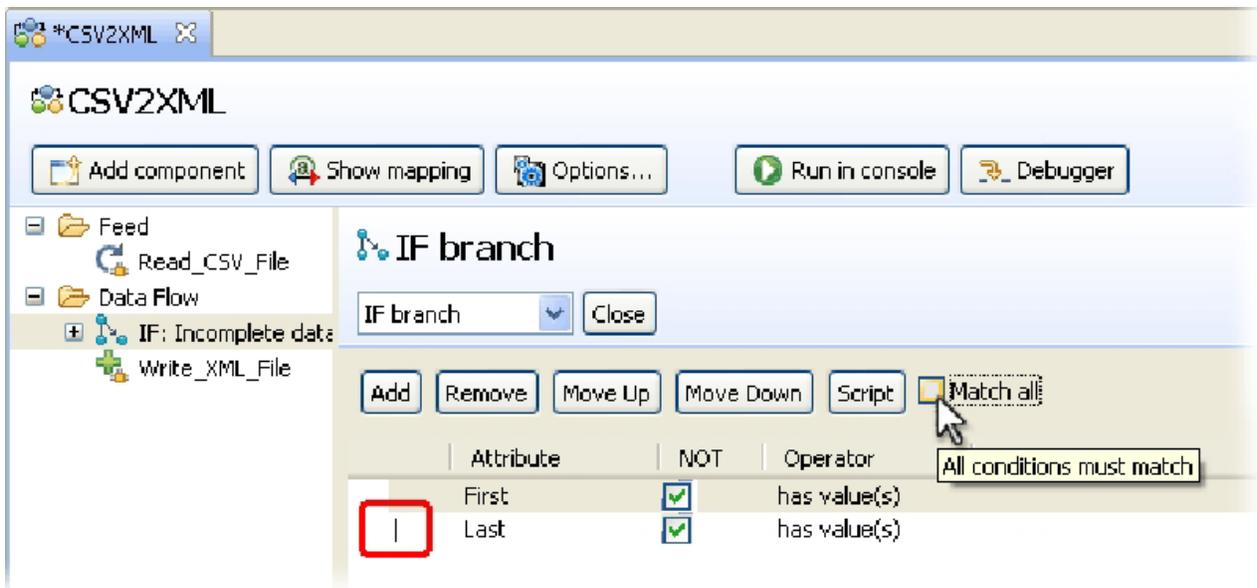


Figura 37. Añadir condiciones simples a la rama IF

Una rama IF desvía el flujo de ejecución de la línea de ensamblaje cuando las condiciones especificadas evalúan a *verdadero*. En su caso, el proceso pasará a los componentes situados bajo esta rama si uno de los atributos 'First' o 'Last' no tiene valores asignados – o incluso si alguno no existe en la entrada de trabajo. Dicho de otro modo, el operador 'tiene el(los) valor(es)' incluye también la comprobación 'existe'.

A continuación, expanda la rama IF y efectúe una doble pulsación en el marcador que aparece debajo de la misma para añadir un componente ahí. En el diálogo **Elegir componente**, pulse el botón de selección **Scripts**, seleccione 'Script vacío' y pulse **Finalizar**. Cambie el nombre de este componente de script (también llamado 'SC') por el de 'Grabar en el registro' e introduzca el siguiente fragmento de código de JavaScript en él²³:

```
task.logmsg("*** Saltando entrada incompleta");
```

La variable *task* que se utiliza aquí hace referencia a la propia línea de ensamblaje y le proporciona una serie de funciones útiles como el método `logmsg()`. Esta llamada en script hace que el mensaje de texto especificado se escriba en la salida del registro.

Para hacer que la salida del registro sea todavía más informativa, incluiremos también los contenidos actuales de la entrada de trabajo. Para hacerlo, pulse el botón derecho del ratón sobre la rama IF, seleccione **Añadir componente...** y marque de nuevo el botón de selección **Scripts**. Esta vez, seleccione el componente de script predefinido 'Volcar entrada de trabajo' ('Dump Work Entry').

23. El editor de scripts proporciona una característica llamada *terminación de código* que le muestra las opciones de que dispone. Por ejemplo, escriba "tas" en el editor de scripts y después pulse la combinación de teclas Ctrl + espacio para abrir el desplegable de terminación de código, que deberá ofrecer una única opción: **task**. Si pulsa Intro, se selecciona esta opción y se escribe en su script. A continuación, escriba la clave punto (.) de forma que en el script ponga "task." y espere un momento. Verá que aparece un nuevo desplegable de terminación de código, esta vez con una lista de todos los métodos y propiedades a los que puede acceder en el objeto *task*.

Finalmente, debe indicar a la LE que detenga el ciclo actual en este punto y devuelva el control al conector iterador para que pueda leer la siguiente línea CSV, excluyendo la Entrada actual de la salida XML. A menos que especifique este comportamiento, el control seguirá al primer componente después de la rama IF. Así pues, debe volver a pulsar el botón derecho del ratón sobre la rama IF, y en **Añadir componente...** seleccionar **Scripts** y después seleccionar el componente de script 'Salir del Flujo'. Ahora su LE debería tener este aspecto:

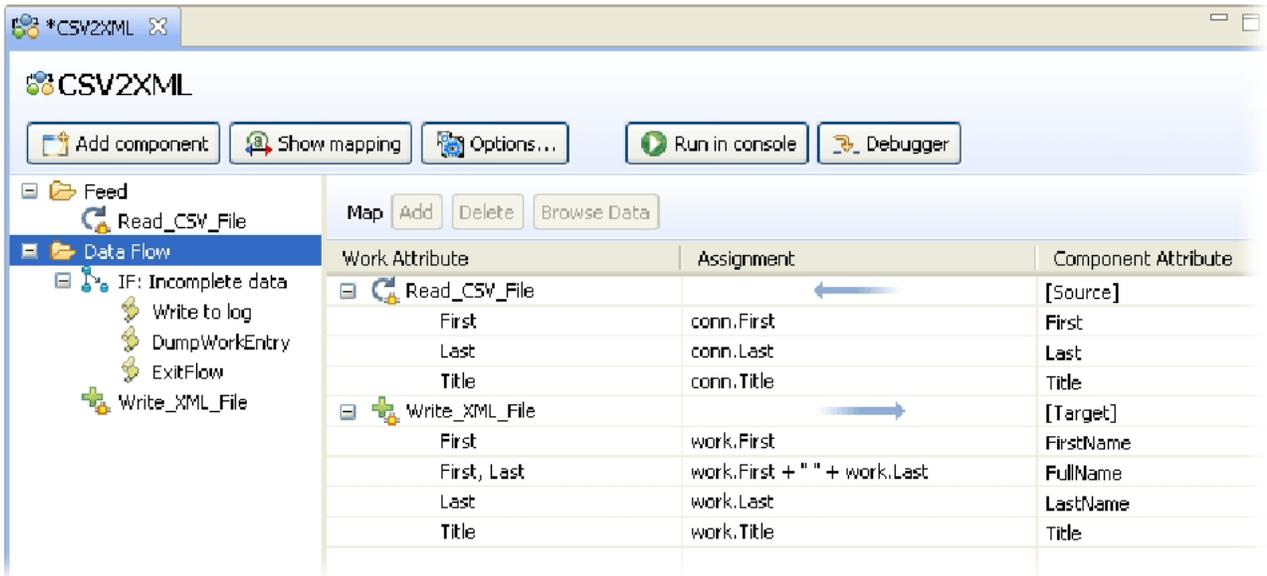


Figura 38. Su primera línea de ensamblaje completa

Antes de volver a ejecutar la línea de ensamblaje, tendrá que abrir el diálogo Comportamiento ante un valor nulo y restaurar las selecciones predeterminadas para la definición de nulo y el comportamiento ante un valor nulo – de lo contrario, las condiciones de la rama IF nunca evaluarán a *verdadero*.

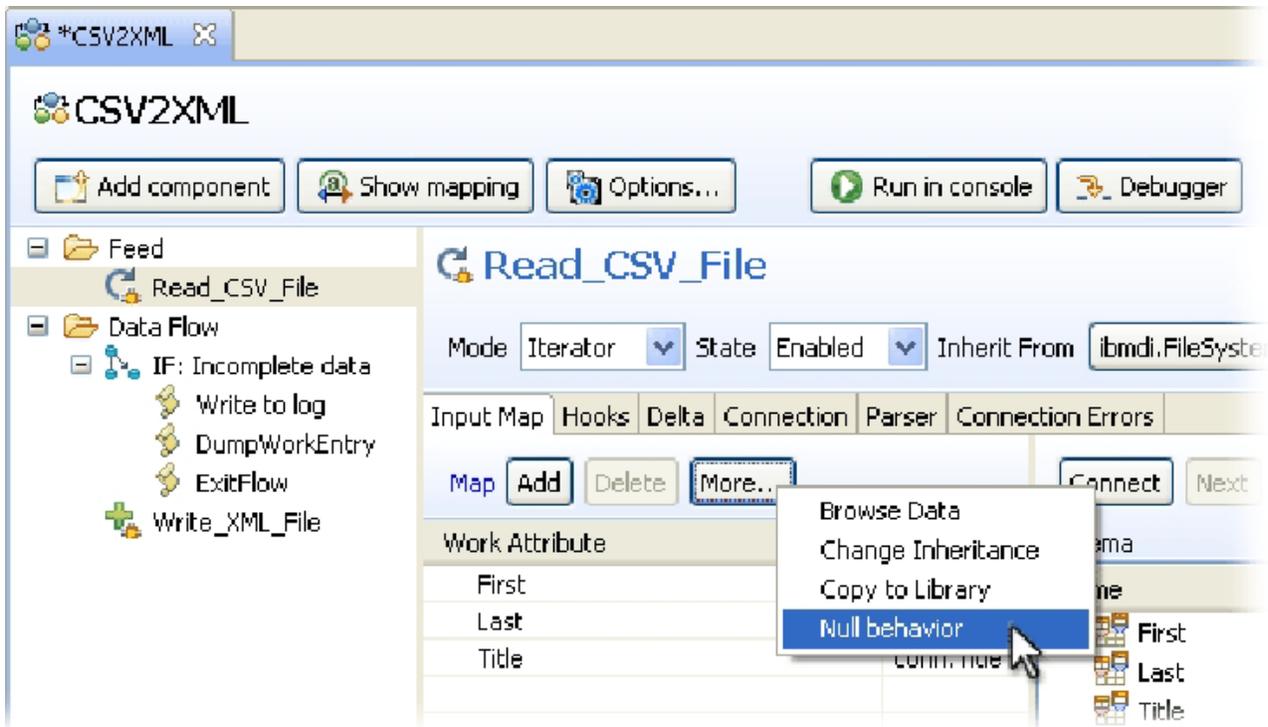


Figura 39. Restablecer el Comportamiento ante un valor nulo de la línea de ensamblaje

Ahora, cuando vuelva a ejecutar la línea de ensamblaje verá su mensaje, seguido del volcado de la entrada de trabajo.

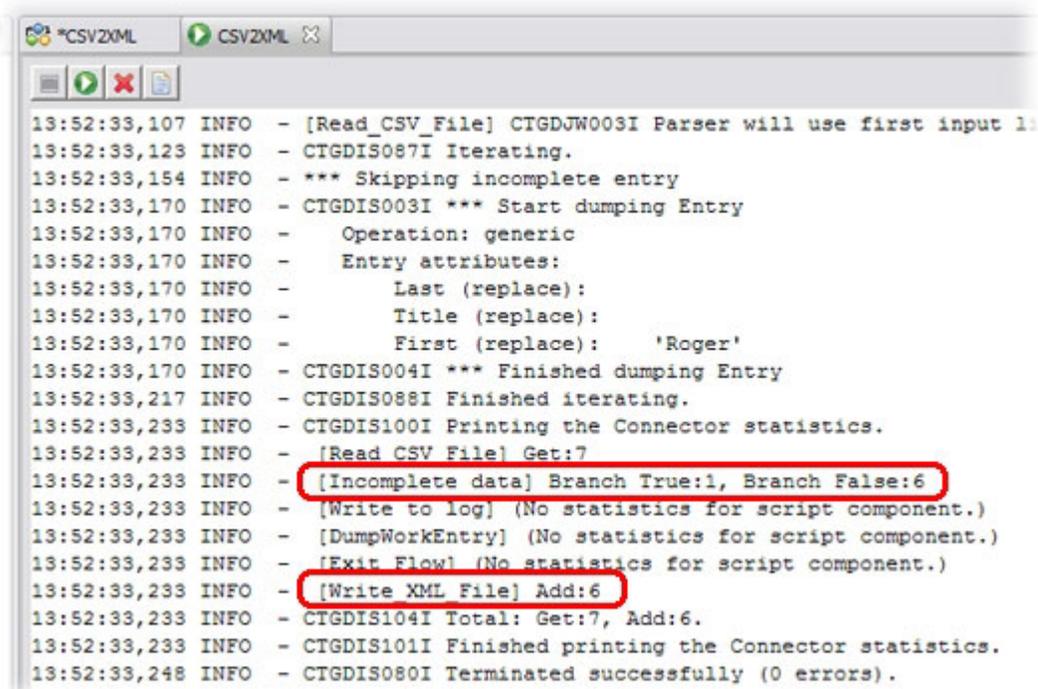


Figura 40. Salida del registro con sus mensajes y el volcado de la entrada de trabajo

En las estadísticas puede ver que la rama IF 'Incomplete Data' ha dado verdadero una vez, con lo que sólo se han añadido seis nodos a la salida XML. Si vuelve a renovar la ventana del Explorador de datos verá que, efectivamente, 'Roger' ha desaparecido.

Depurar la línea de ensamblaje

Una de las características más potentes de IBM Security Directory Integrator es su depurador integrado de líneas de ensamblaje, que le permite ejecutar paso a paso la línea de ensamblaje para ver e incluso modificar los datos al vuelo.

Para ejecutar paso a paso su primera línea de ensamblaje, pulse el botón **Iniciar sesión de depuración**.

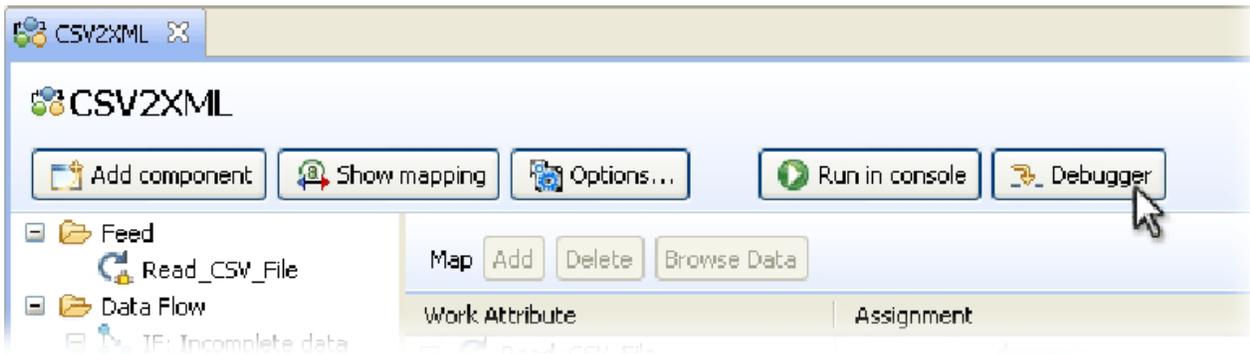


Figura 41. Depurar la línea de ensamblaje

En vez de ver la ventana estándar Ejecutar consola, se encontrará en el depurador.

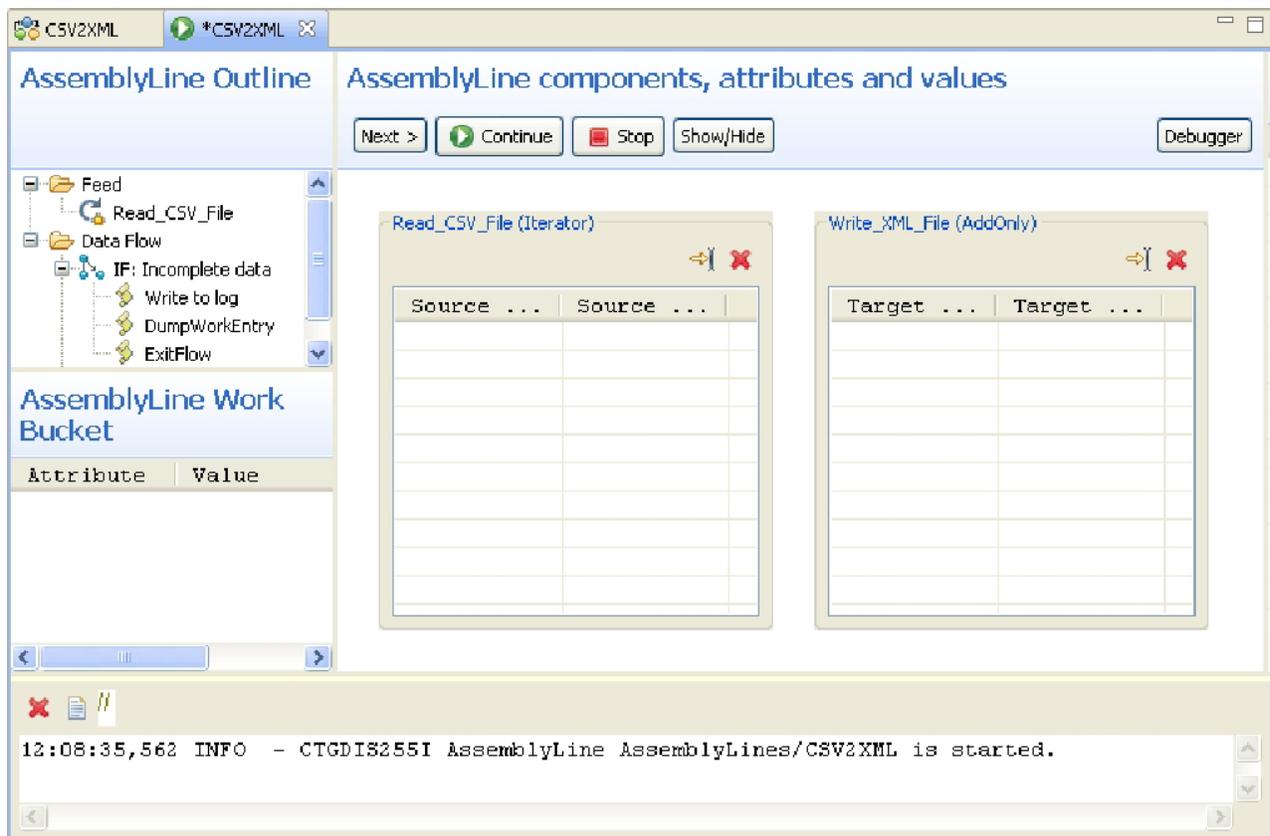


Figura 42. El repetidor de pasos de la línea de ensamblaje

El Depurador de línea de ensamblaje ofrece dos modalidades: el repetidor de datos, que proporciona características de prueba simples y directas, y el depurador de línea de ensamblaje avanzado, en el que puede detallar más, como ejecutar paso a paso los scripts, trabajar interactivamente con las bibliotecas Java y modificar los datos en el archivo.

El repetidor de datos es una herramienta útil para realizar paso a paso la ejecución de los conectores de línea de ensamblaje y ver los datos leídos, escritos y transformados. Esta pantalla está dividida en tres áreas principales:

- El **Resumen de la línea de ensamblaje** muestra su LE y resalta dónde se detiene la ejecución;
- El **Contenedor de trabajo de la línea de ensamblaje** que muestra todos los atributos correlacionados en la entrada de trabajo, es decir, las encontradas en Correlaciones de entrada o Componentes de una correlación de atributos.
- **Componentes de línea de ensamblaje, Atributos y Valores** donde hay una fila de botones para controlar la sesión de depuración y un conjunto de datos muestra cuadrículas para todos los conectores en la LE.

En la parte inferior puede ver una ventana de salida de la consola que muestra la misma información que obtuvo al ejecutar la línea de ensamblaje utilizando el botón Ejecutar.

Llegado este punto, la línea de ensamblaje se ha asignado al servidor de pruebas y está preparada para empezar a ejecutarse cuando usted decida. Pulse el botón Siguiente para empezar a realizar los pasos.

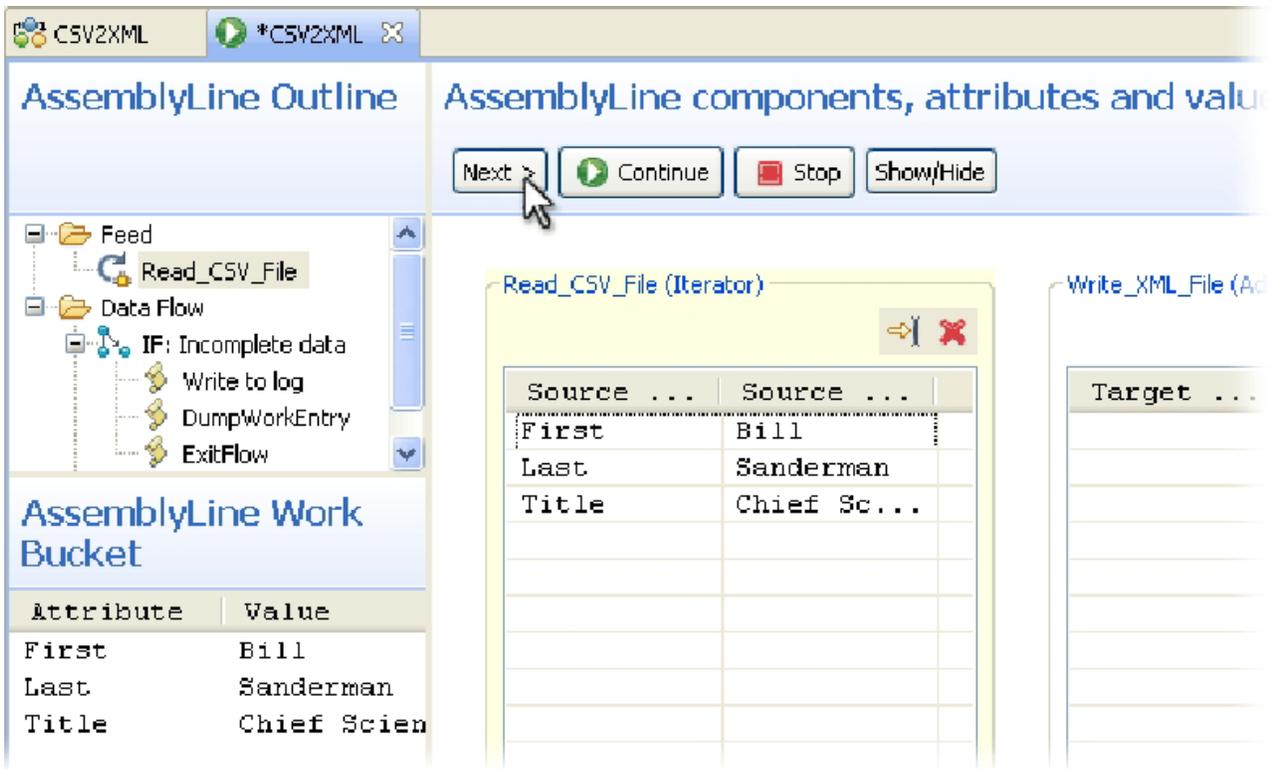


Figura 43. Realización de pasos de la ejecución de LE

Observe cómo ocurren en la pantalla estas tres cosas: el **Resumen de la línea de ensamblaje** muestra que el conector "Read_CSV_File" está activo actualmente, el **Contenedor de trabajo de la línea de ensamblaje** muestra los atributos que el conector acaba de leer, y la cuadrícula de visualización de datos de este conector también está llena de estos atributos. Cada vez que pulsa el botón **Siguiente** la ejecución continúa en el paso siguiente y se renuevan las pantallas de información.

También puede utilizar el botón **Ejecutar hasta aquí**, situado en la parte superior de la cuadrícula de datos, para que un conector salte a este lugar. Haga esto ahora para el conector "Write_XML_File".

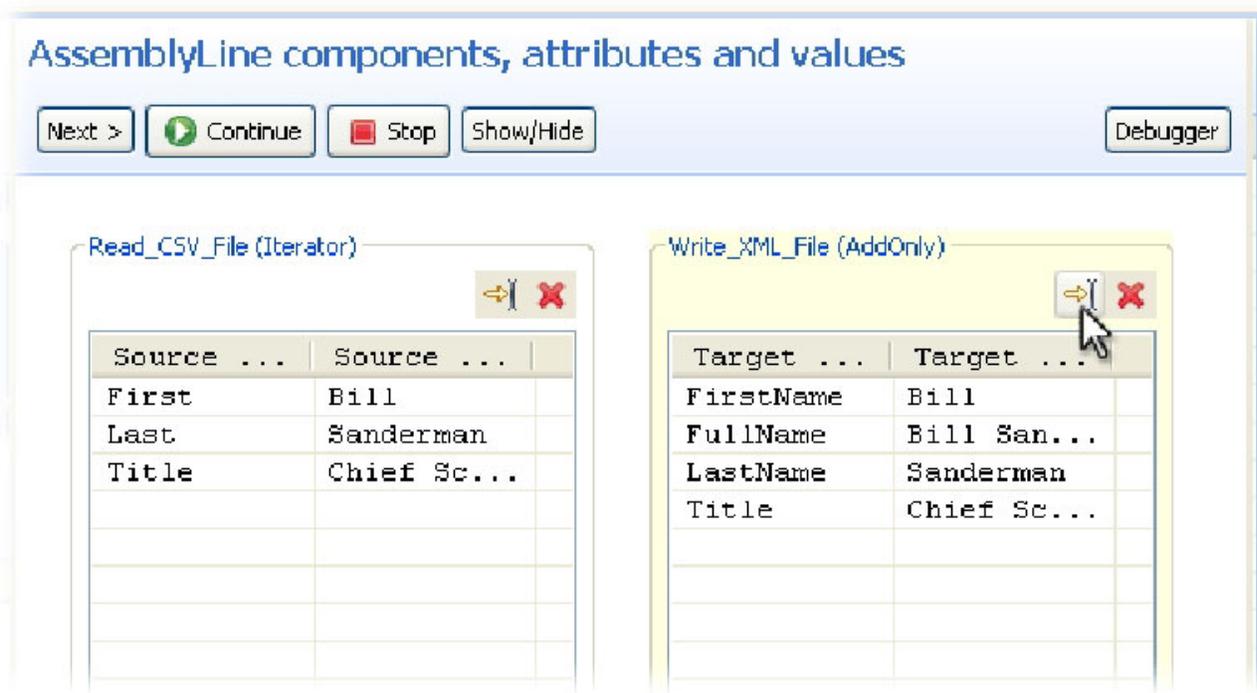


Figura 44. Realización de pasos del conector Write_XML_File

Esta cuadrícula de datos muestra los atributos en la correlación de salida de este conector, así como sus valores, incluido el valor calculado para "FullName".

Ahora echemos un vistazo a los botones de la barra de herramientas del repetidor de datos para ver qué opciones tiene:

- **Siguiente** > mueve el proceso al siguiente componente y actualiza todas las áreas de visualización de datos
- **Continuar** hace que la línea de ensamblaje continúe la ejecución hasta que termine
- **Detener** termina la ejecución inmediatamente
- **Mostrar/Ocultar** permite decidir qué cuadrículas de datos de conector se deben mostrar
- **Depurador** cambia a la modalidad Depurador completo, donde puede ejecutar paso a paso el código de script, establecer puntos de interrupción, ver y modificar variables de atributos y de script, ejecutar de forma interactiva mandatos JavaScript en el contexto de la línea de ensamblaje en ejecución.

Aunque el repetidor de datos proporciona mucha información sobre qué rendimiento tendrá la línea de ensamblaje, a veces es necesaria la eficacia añadida del depurador avanzado. Tenga en cuenta que puede alternar entre la modalidad de repetidor y la modalidad avanzada con la frecuencia que desee durante una sesión de depuración. Inténtelo ahora pulsando el botón **Depurador**, situado en el extremo derecho de la fila de botones Repetidor de datos.

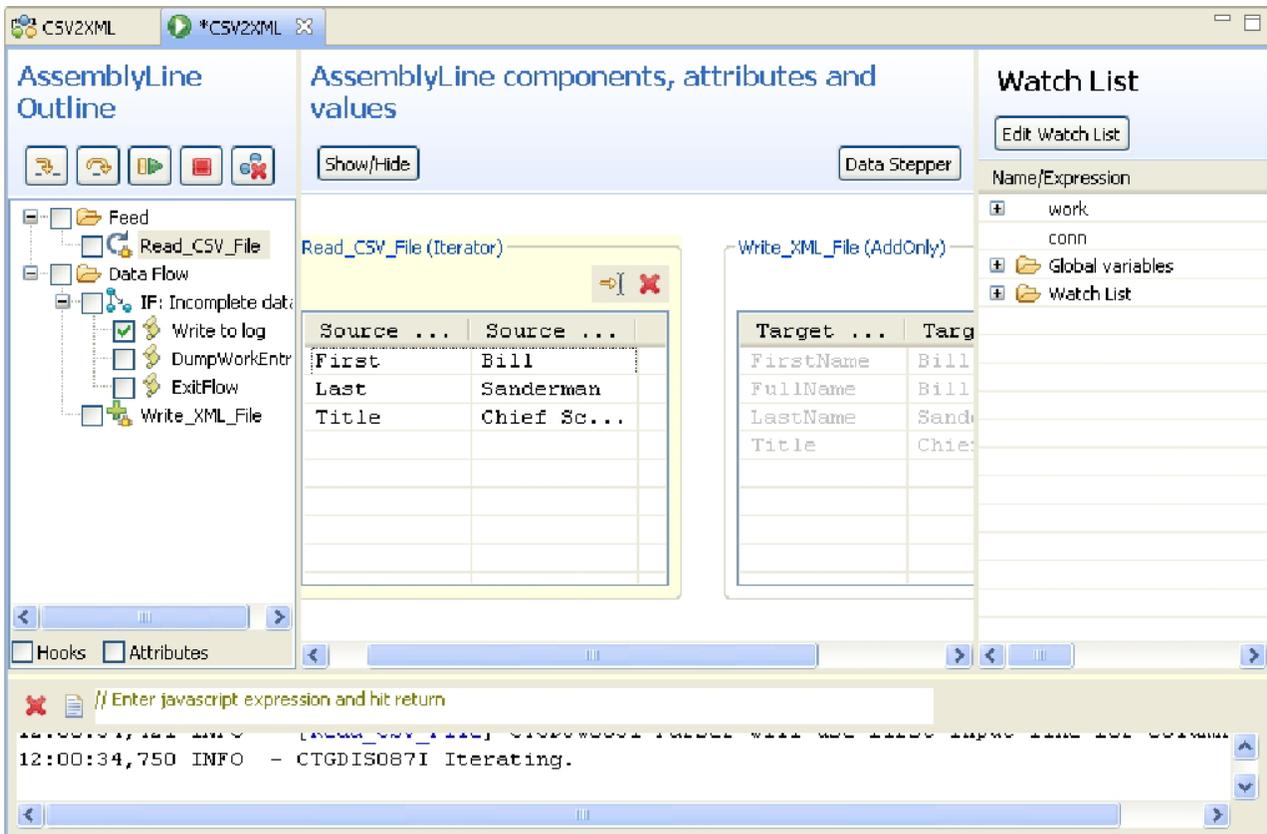


Figura 45. Modalidad de depurador avanzado

Al alternar las modalidades, la pantalla se vuelve a configurar para proporcionar nuevos controles para el **Resumen de la línea de ensamblaje** y el **Contenedor de trabajo de la línea de ensamblaje** se sustituye por la **Lista de vigilancia** en la parte derecha de la ventana. La **Lista de vigilancia** muestra el atributo estándar *contenedores*: work y conn. También hay una carpeta llamada "Variables globales" que si se abre muestra todas las variables definidas para la línea de ensamblaje; tanto las incorporadas como **work** y **system**, como para las que defina en el código de script. La última carpeta Watch es para su propio uso y puede añadir variables o expresiones completas de JavaScript que desee observar utilizando el botón **Editar lista de vigilancia** en la parte superior de este panel.

Si centramos nuestra atención en el Resumen de la **línea de ensamblaje**, los recuadros situados junto a los componentes de esta vista de árbol se denominan Puntos de interrupción, y puede indicar a IBM Security Directory Integrator que se detenga en cualquier componente durante la ejecución pulsando en uno de estos. También puede pulsar el botón derecho del ratón en cualquier nodo y seleccionar **Ejecutar e interrumpir aquí** para realizar la ejecución de LE en este punto. La barra de herramientas anterior al resumen le ofrece algunos de los controles que tenía en el repetidor de datos, además de un par de controles nuevos:



Figura 46. Botones del depurador

Empezando por la izquierda, estos botones son:

- **Ejecutar paso a paso**, que le permite ejecutar paso a paso las correlaciones de atributos, los scripts e incluso los flujos de trabajo subyacentes de la línea de ensamblaje y sus componentes. Estos puntos de paso en los flujos incorporados se denominan "*Enganches*" y se describen en un ejercicio posterior
- **Saltar paso**, que es igual que el botón **Siguiente**> que vio en el repetidor de datos. En el depurador también le permite detenerse en las llamadas de función de script en lugar de dentro de ellas
- **Continuar**, hace que la línea de ensamblaje se ejecute hasta que se complete (de igual modo que en el repetidor de datos) o hasta que se alcance un punto de interrupción
- **Detener**, detiene la línea de ensamblaje, del mismo modo que en el repetidor de datos
- **Borrar todos los puntos de interrupción**, elimina los puntos de interrupción que establecidos para la línea de ensamblaje.

Para familiarizarse con cómo funcionan los puntos de interrupción, intente establecer uno para el script "Grabar en el registro" pulsando en el recuadro que hay junto a este componente.

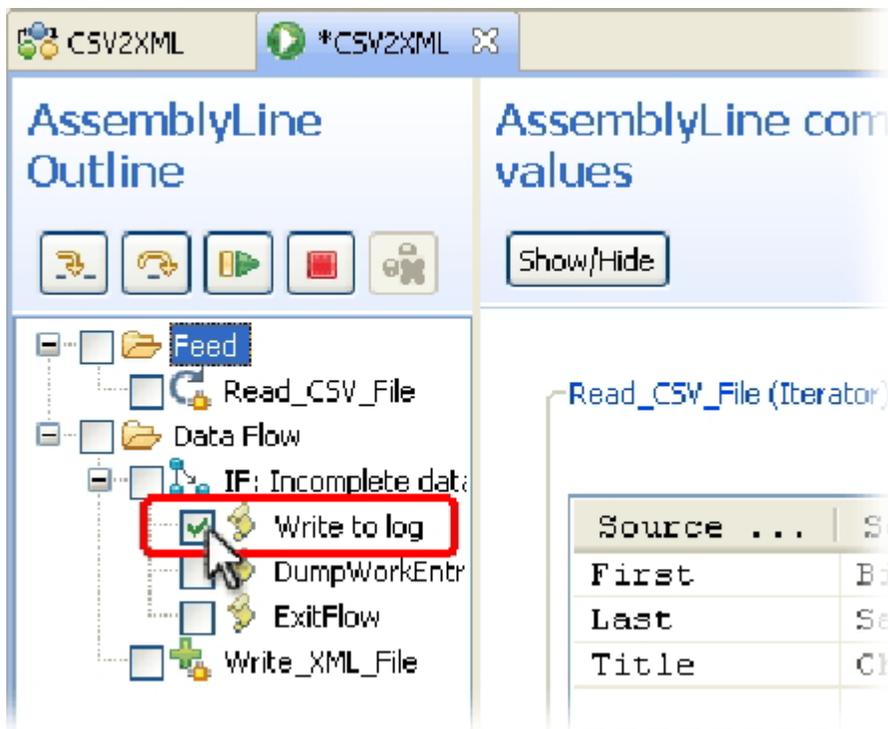


Figura 47. Establecimiento de un punto de interrupción

Ahora pulse el botón **Continuar** y la línea de ensamblaje se ejecutará hasta que la rama IF sea *verdadera* y encuentre el control en el script "Grabar en el registro". IBM Security Directory Integrator también abre un aquí el área Script que le permite ejecutar paso a paso el código. Puede incluso establecer puntos de interrupción en cualquier línea de script realizando una doble pulsación en el margen que está a la izquierda de esa línea.

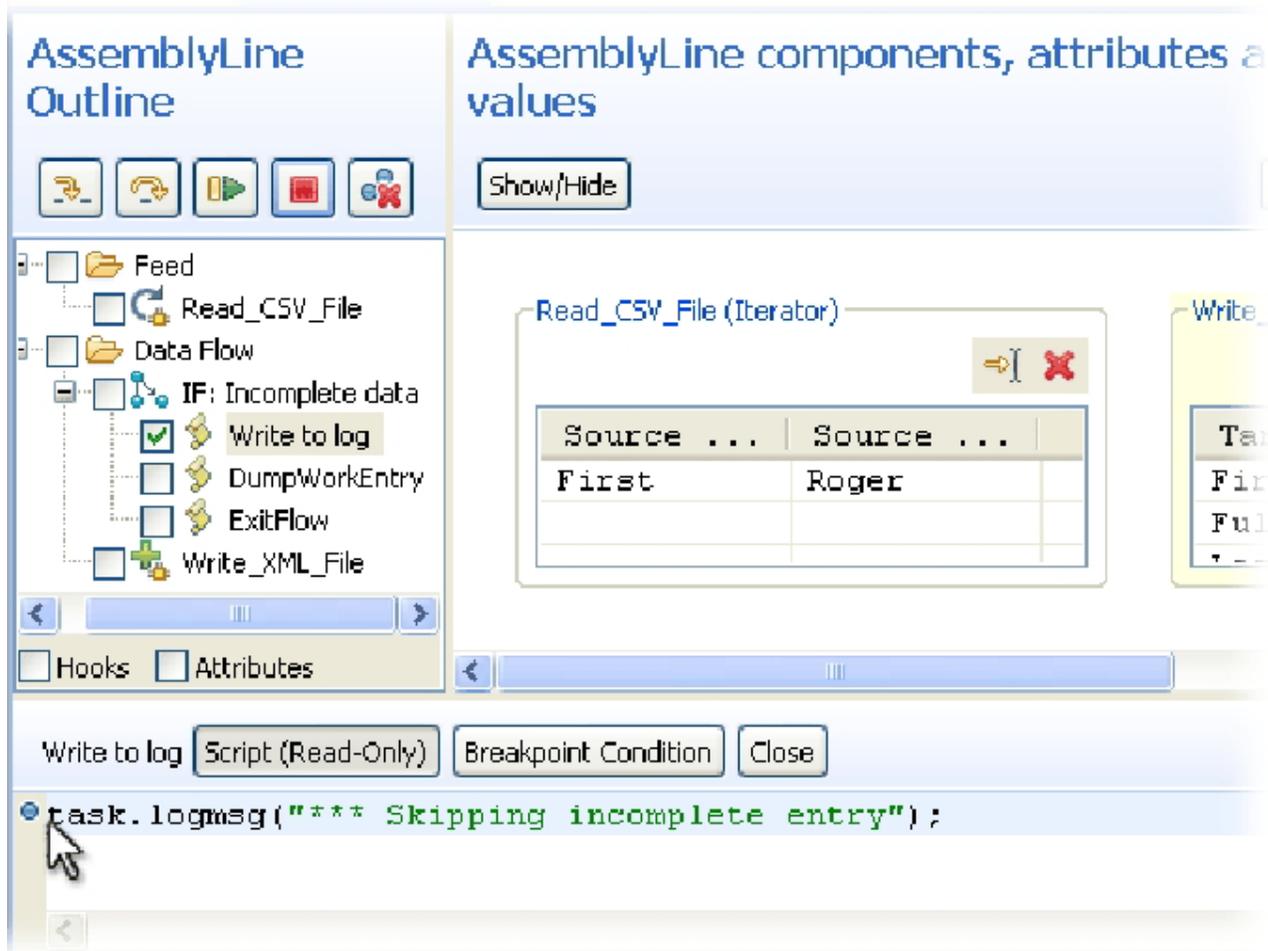


Figura 48. Establecimiento de un punto de interrupción en un script

Asimismo, puede efectuar una doble pulsación en cualquier nodo de la lista de componentes para abrir la pantalla del depurador. Como puede ver en la figura anterior, hay un botón llamado **Condición de punto de interrupción**. Puede utilizar este botón para establecer una expresión JavaScript que debe evaluarse como *true* o *false* y que determinará si un punto de interrupción es o no activo. Por ejemplo, el punto de interrupción mostrado antes se puede establecer como verdadero si:

```
work.First.startsWith("R")
```

o bien

```
mycounter > 1000
```

Esto es muy útil para depurar problemas que sólo se producen en algunos conjuntos de datos de entrada.

Y si por alguna razón necesita volver a un paso anterior, simplemente detenga y vuelva a iniciar la sesión de depuración. También puede volver al repetidor de datos pulsando el botón Repetidor de datos.

Pero antes de cerrar el depurador avanzado, hay una característica más que vale la pena considerar: la **línea de mandatos de evaluación JavaScript**.

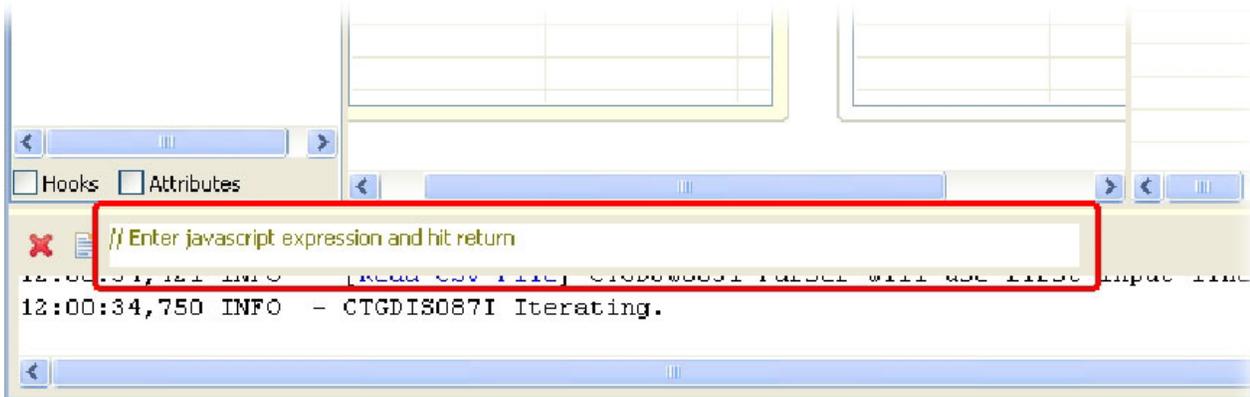


Figura 49. Línea de mandatos de evaluación JavaScript

Este campo de entrada de aspecto inocente que está encima del área de salida de registro le permite ejecutar un fragmento de código de script en el contexto de la línea de ensamblaje en ejecución. Pruébalo ahora escribiendo este mandato y pulsando Intro:

```
task.dumpEntry(work)
```

Este mandato hará que se muestre el contenido de la entrada de trabajo en la ventana de salida de registro. Ahora, introduzca estos datos:

```
i = 42
```

Verá aparecer el siguiente mensaje en el registro: `i=42 >> 42.0`

Esto le indica que ha definido una nueva variable ('i') con el valor 42. La propia expresión se evalúa (al igual que todas las sentencias de script) en el valor de la asignación. También puede cambiar los valores de la variable y los atributos ya definidos en la LE, por ejemplo:

```
work.First="Rudy"
```

Después de ejecutar esta línea, el valor del atributo "First" será "Rudy". La capacidad de modificar los datos directamente significa que puede asegurarse de que la línea de ensamblaje ejecuta paso a paso la lógica de rama, lo que le permite probar totalmente la solución.

Le recomendamos que invierta algún tiempo en familiarizarse con el depurador y el repetidor de pasos de línea de ensamblaje. No sólo le ayuda a comprender mejor el funcionamiento de la LE, incluyendo todos los flujos de trabajo incorporados, proporcionados por el kernel del servidor de IBM Security Directory Integrator, sino que también le ayudará a validar su propia implementación y las suposiciones sobre sus datos.

Buscar datos en un origen de datos secuencial

Siguiendo con el caso de ejemplo utilizado para el aprendizaje, el siguiente paso es añadir la búsqueda en D2.

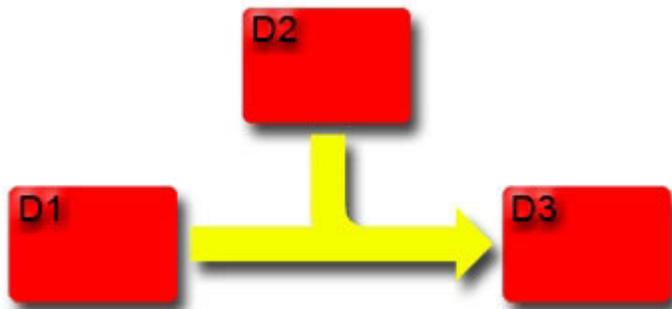


Figura 50. Diagrama de flujo del caso de ejemplo

El directorio Tutorial contiene un archivo denominado PhoneNumbers.xml que utilizaremos como el origen de datos D2. Este archivo contiene una serie de entradas XML, cada una con dos atributos: 'User' y 'telephoneNo'.

Su siguiente tarea será incluir 'telephoneNo' como parte de los datos a escribir en el documento XML de salida. Puesto que no puede acceder aleatoriamente a este archivo de texto para realizar una búsqueda, como haría en una base de datos o en un directorio, deberá encontrar el número de teléfono correcto para cada entrada CSV recorriendo el archivo en bucle y comparando el atributo 'User' del archivo con el atributo 'FullName' procedente de la entrada CSV actual.

Sin embargo, 'FullName' se calcula en la Correlación de salida del conector 'Write_XML_File' – es decir, demasiado tarde para realizar la comparación. Esto significa que debe mover este atributo calculado desde la Correlación de salida 'Write_XML_File' a la Correlación de entrada 'Read_CSV_File'. Para hacerlo, arrastre el elemento de correlación de atributos de la correlación de salida a la de entrada.

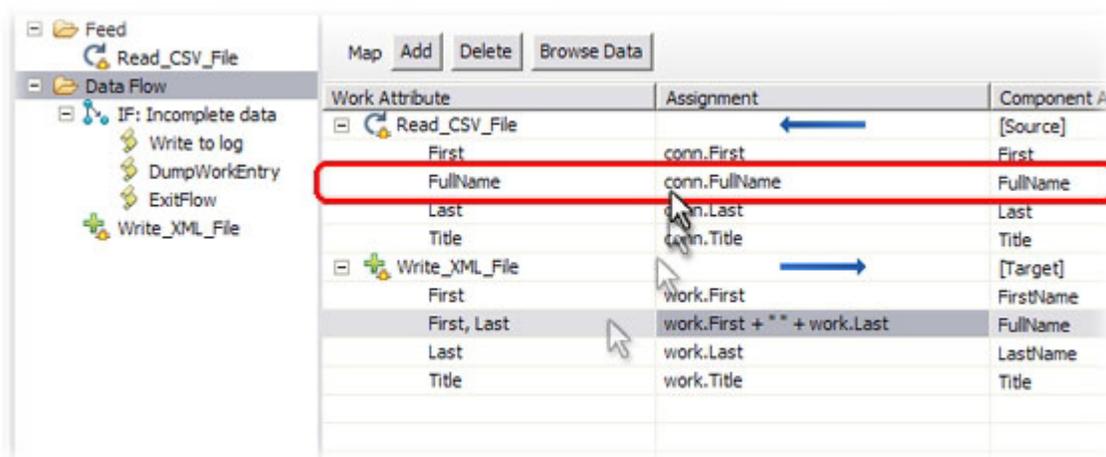


Figura 51. Arrastrar 'FullName' a la correlación de entrada del conector iterador

Ahora habrá un elemento 'FullName' en ambas correlaciones. Deberá ajustar la asignación de la correlación de entrada, puesto que ahora está correlacionando desde la entrada de conector a la entrada de trabajo y no al revés, como se hace en la Correlación de salida. Para hacerlo, efectúe una doble pulsación sobre 'FullName' bajo 'Read_CSV_File' y cambie la asignación a: `conn.First + " " + conn.Last`

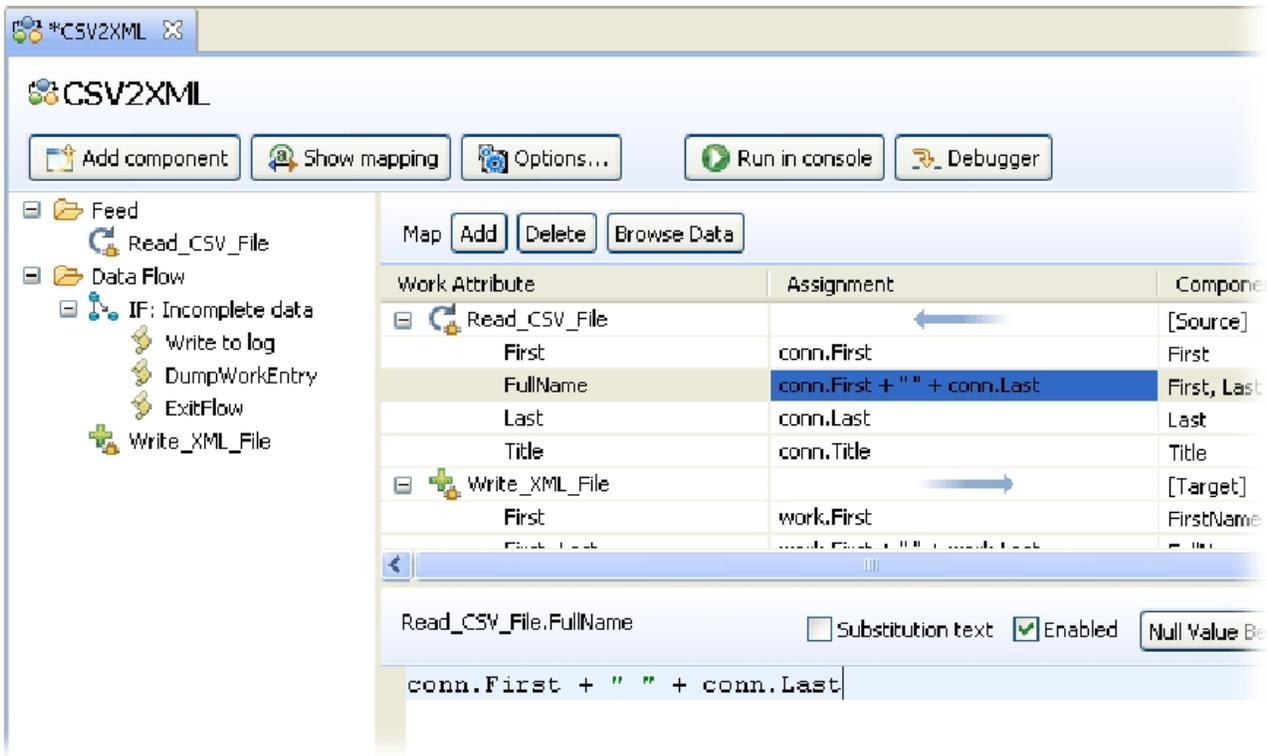


Figura 52. Editar la asignación de 'FullName'

También puede editar el elemento original 'FullName' cambiando su asignación a, sencillamente, work.FullName puesto que este atributo ahora estará disponible en la entrada de trabajo, gracias a la correlación de entrada que ha modificado.

A continuación, vuelva a ejecutar la línea de ensamblaje y compruebe que el archivo Output.xml no se ha modificado. Una vez haya confirmado esto, utilizará un componente de *Bucle* para leer el archivo PhoneNumbers.xml y buscar el número de teléfono de cada usuario²⁴.

Empiece por añadir un nuevo componente a la sección **Flujo de datos**, seleccionando esta vez el componente denominado *Bucle de conector* y renombrándolo a 'TelephoneNumber'. Un Bucle de conector es un componente de bucle que utiliza un conector para leer información de un origen de datos y después recorre en ciclos todos los componentes que tiene adjuntos una vez para cada entrada que devuelva el Conector. Es similar al comportamiento *for-each* de un conector iterador de la sección Canal de información, que recorre en ciclos los componentes de la sección **Flujo de datos** para cada entrada leída.

Arrastre el nuevo bucle de conector 'TelephoneNumber' hasta colocarlo entre la rama IF y el conector 'Write_XML_File'. Asegúrese de que no queda *dentro* de la rama IF.

24. Hay tres tipos de componentes de Bucle: 1) El *Bucle de conector*, que le permite recorrer en ciclos los datos que devuelve un conector en modalidad iterador o en modalidad búsqueda. Éste es el tipo de bucle que utilizará en este ejercicio; 2) el *Bucle para cada valor de atributo*, que le permite recorrer en ciclos los valores de un Atributo multivalor, como los que se encuentran en sistemas como Lotus Notes y directorios LDAP; y 3) el *Bucle condicional*, que utiliza Condiciones simples y en script - como las utilizadas por las ramas - para controlar el recorrido en ciclos

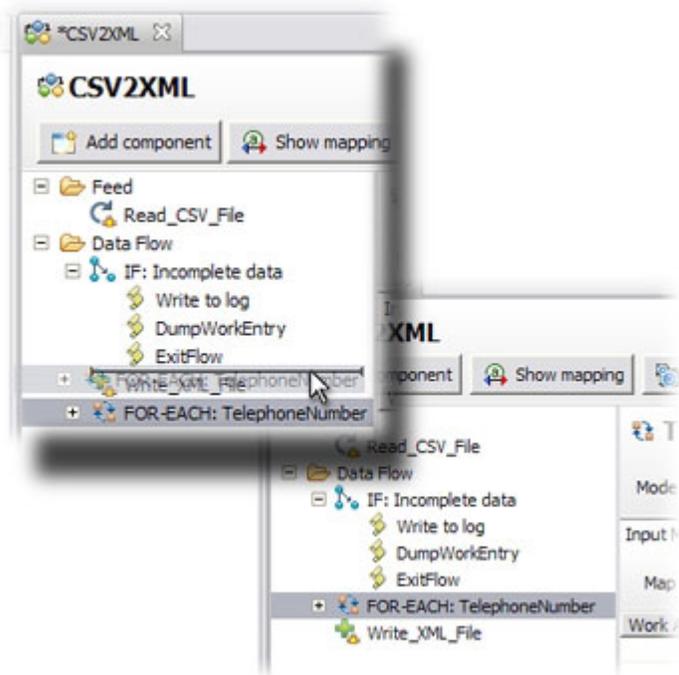


Figura 53. Arrastrar el bucle de conector

A continuación, selecciónelo para abrir su editor, que es similar a un editor de conector.

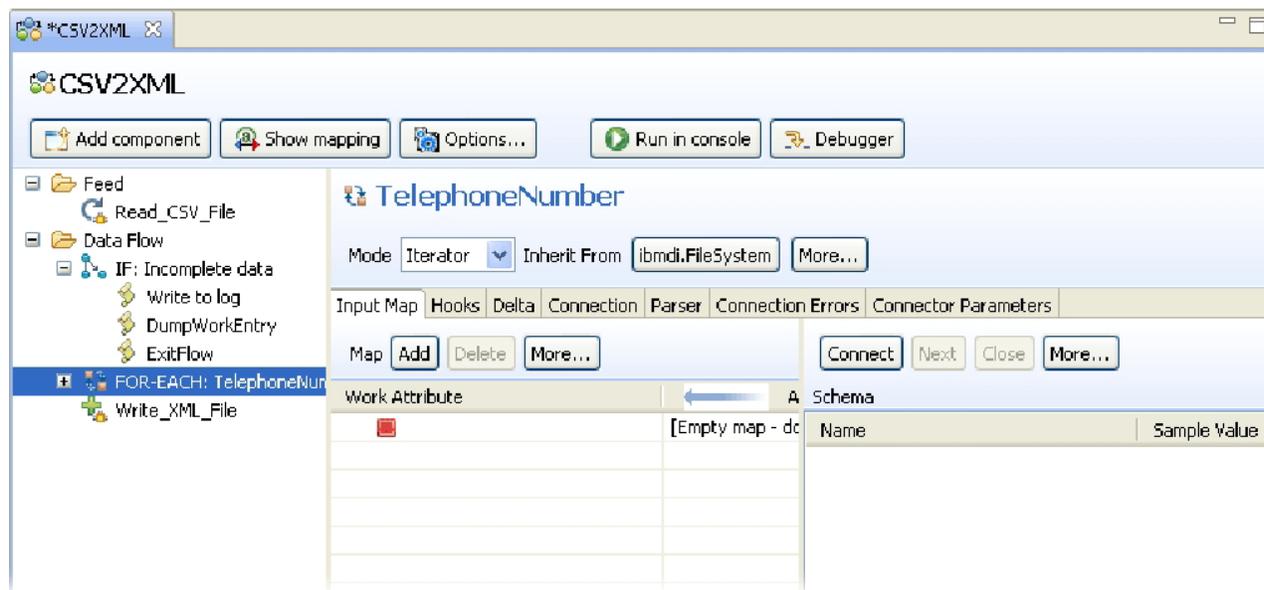


Figura 54. Configuración del bucle de conector

Las diferencias principales son que el desplegable **Modalidad** sólo contiene las opciones **Iterador** y **Buscar**. Además, hay un botón **Más...** que proporciona opciones para limitar las entradas que se recorren en ciclo, así como un parámetro

desplegable **Inicializar** con las tres opciones siguientes:

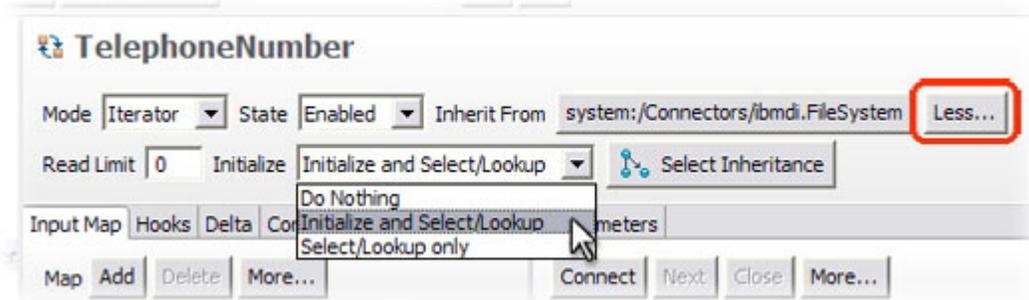


Figura 55. Valores avanzados del bucle de conector

- **No hacer nada**, que significa que cuando se llega a este bucle durante el proceso de la LE, no se inicializará el conector que se incluye en el mismo.
- **Inicializar y seleccionar/buscar**, para hacer que el conector se inicialice cuando el bucle empiece a recorrer las entradas en ciclo. Utilice esta opción para el ejercicio actual, ya que su bucle de componente leerá de un archivo y desea que empiece desde el principio cada vez.
- **Sólo seleccionar/buscar**, para cuando el bucle de conector apunta a una base de datos, un directorio o algún otro origen de datos accesible aleatoriamente. En este caso no es necesario reinicializar la conexión cada vez. Es suficiente con reiniciar la búsqueda, que es una operación de *Seleccionar* en el caso de la modalidad Iterador, y *buscar* en el caso de la modalidad Buscar.

Configure el conector de bucle (que, de forma predeterminada es del tipo 'Sistemadearchivos') para que lea el archivo `PhoneNumber.xml` y después seleccione el 'Analizador XML'. A continuación, abra la pestaña Correlación de entrada para descubrir los atributos.

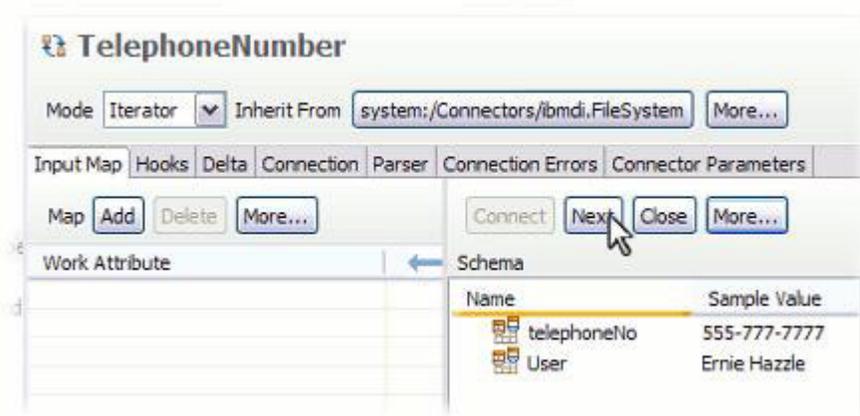


Figura 56. Atributos jerárquicos

Para realizar la correlación a nivel de atributos, seleccione 'User' y 'telephoneNo' del esquema de entrada y arrástrelos a la correlación de entrada.

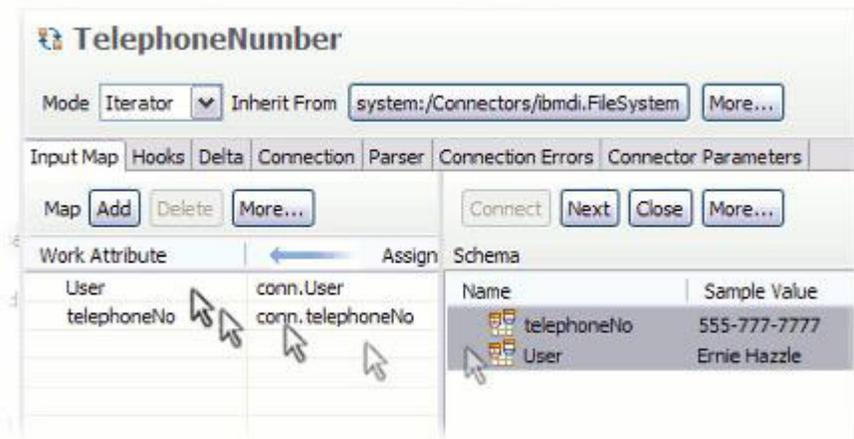


Figura 57. Arrastrar desde el Esquema hasta la Correlación de entrada

Cuando lo haya hecho, tendrá una regla de correlación para un atributo denominado "User" y una para "telephoneNo".

Ahora ya puede cerrar el editor del conector de bucle y añadir una rama IF bajo la misma pulsando con el botón derecho del ratón sobre el bucle de conector y seleccionando **Añadir Componente...**. Como nombre de esta nueva rama IF escriba 'Matching name found'. Y añada una condición simple para comprobar si 'User' equivale a '\$FullName'²⁵.

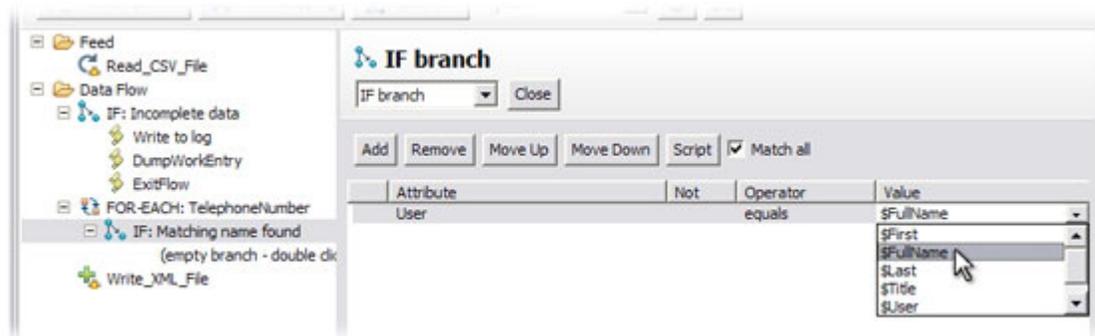


Figura 58. Editor de condiciones para la rama IF

El comportamiento deseado es que, cuando se encuentre una coincidencia, el Bucle de conector salga del bucle con los valores correctos en los atributos 'User' y 'telephoneNo'. Para conseguirlo, añada un componente de script con el nombre 'Exit loop' y escriba el siguiente script dentro del mismo:

```
system.exitBranch("loop");
```

Pero, ¿qué ocurre si el bucle de conector llega al final del archivo PhoneNumbers.xml sin encontrar ninguna coincidencia? Los atributos 'User' y 'telephoneNo' contienen los valores obtenidos de la última entrada del archivo, con lo cual, en este caso no sirve comprobar si los atributos están vacíos. Tendrá que idear alguna otra forma para detectar que no ha habido ninguna coincidencia.

25. El signo dólar es un carácter especial que se utiliza aquí para indicar que 'FullName' no es una serie literal que debe coincidir, sino el valor de un atributo que se encuentra en la entrada de trabajo.

La respuesta es utilizar una variable de script como distintivo para indicar que se ha encontrado una coincidencia. Para hacerlo, añada un componente de script llamado 'Found user' dentro de la rama IF, y arrástrelo hasta la posición justo antes del componente de script 'Exit loop'. Este componente de script debe contener el siguiente fragmento de código:

```
foundUser = true;
```

Para indicar que se ha llegado al final del archivo de entrada sin encontrar ninguna coincidencia, seleccione el bucle de conector y abra la pestaña Enganches.

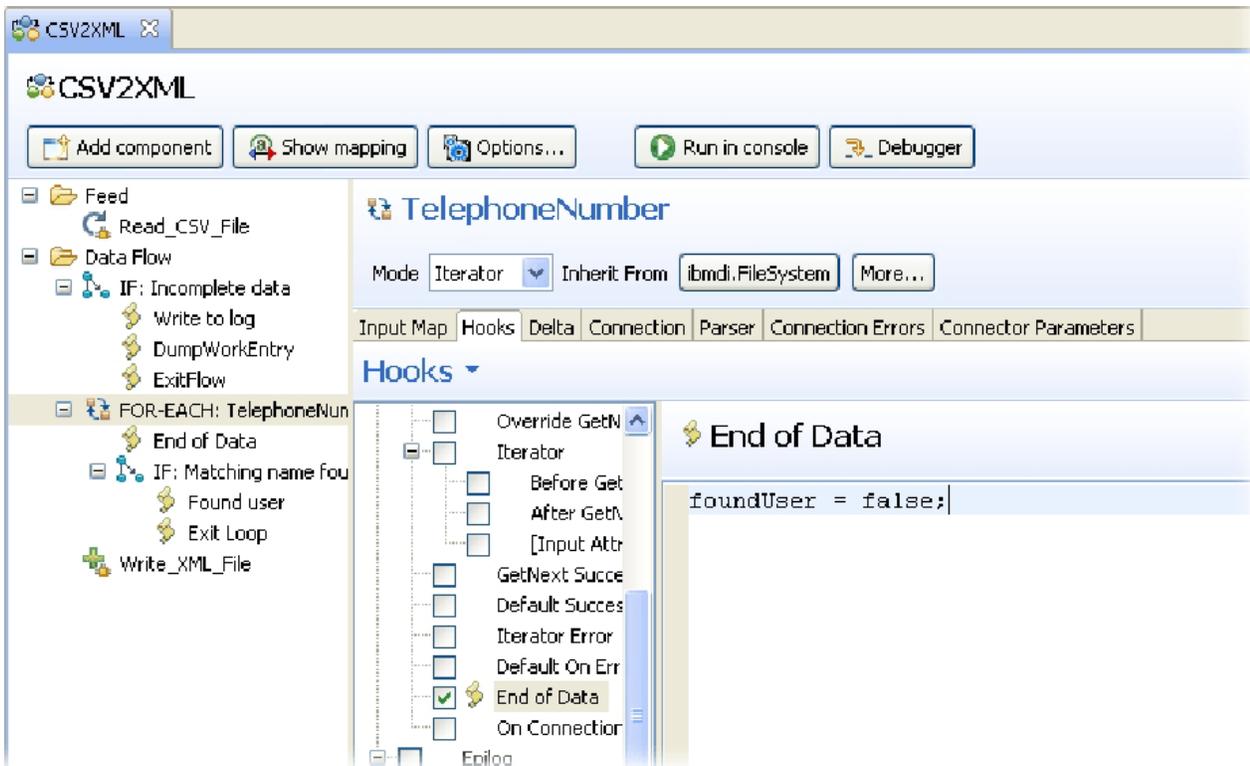


Figura 59. Escribir el script para el enganche Fin de los datos

Seleccione el enganche denominado 'Fin de los datos' y escriba el siguiente código script.

```
foundUser = false;
```

Sólo se llegará al enganche 'Fin de los datos' si el conector intenta leer más allá de la última entrada en el origen de datos conectado. Si esto ocurre, es que no se ha encontrado ninguna coincidencia.

Ahora su línea de ensamblaje debería tener los siguientes componentes:

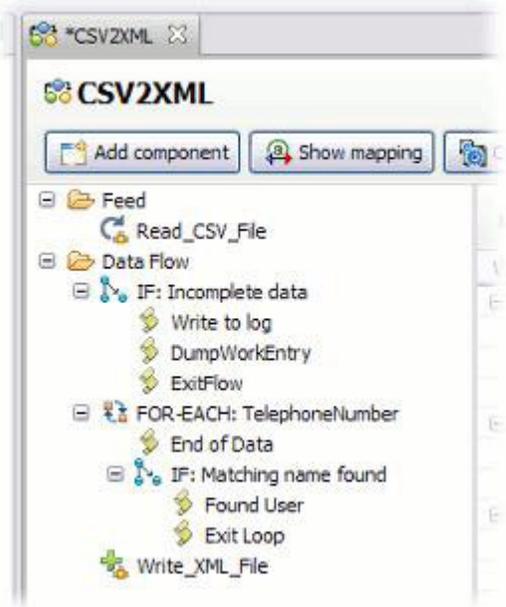


Figura 60. Lista de componentes de la sección Flujo de Datos de la línea de ensamblaje

Ahora ya debería poder decir si la búsqueda ha sido o no satisfactoria comprobando la variable de script. Esto es importante porque, para cuando no se encuentra ninguna coincidencia, debe definir también un valor predeterminado para el atributo 'telephoneNo'; de lo contrario mantendrá el último valor que haya leído el bucle de conector.

Así pues, cree otra rama IF inmediatamente después del bucle de conector i póngale el nombre 'NOT foundUser'. Pulse el botón **Script** en el área de detalles de la **Rama IF** y escriba el siguiente script para comprobar el valor de la variable de script:

```
! foundUser
```

El signo de exclamación niega el valor de foundUser, de forma que si se ha establecido a *falso* en el enganche 'Fin de los datos' del bucle de conector, esta condición de rama evaluará a *verdadero*.

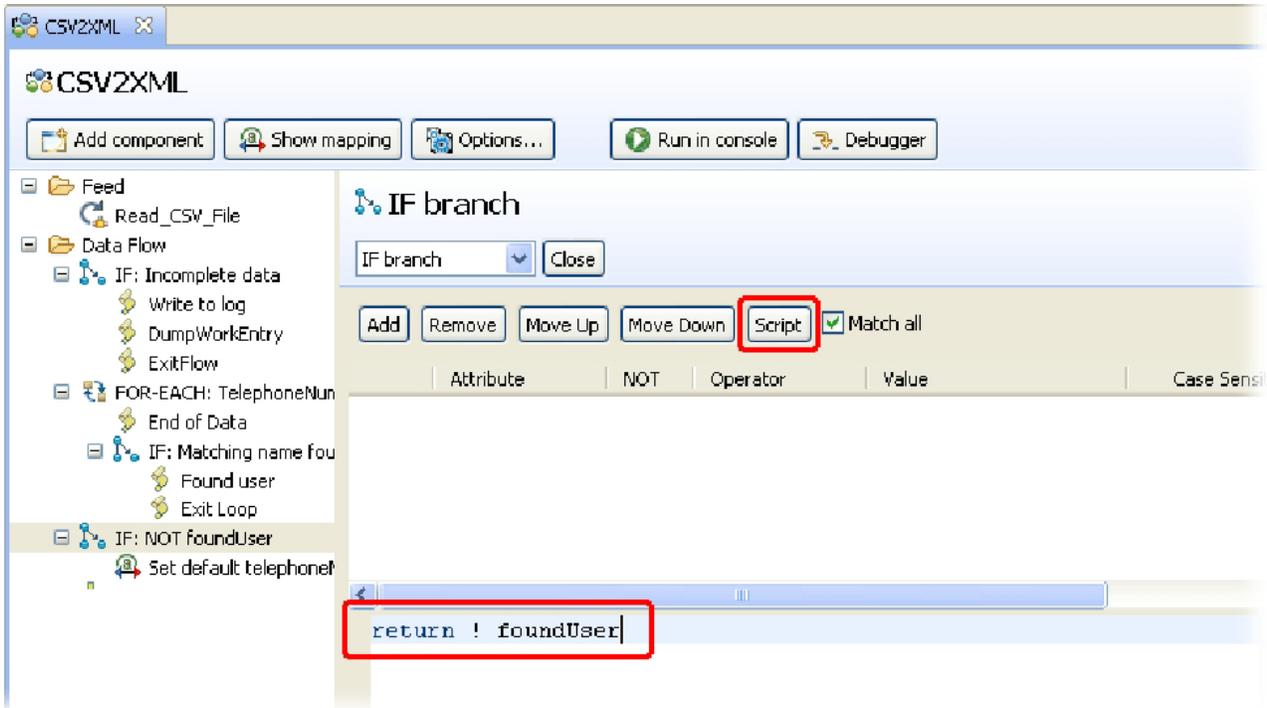


Figura 61. Escribir un script de condición para la rama IF

Añada un nuevo componente de tipo 'Correlación de atributos' bajo la misma. Como nombre de este componente de Correlación de atributos, escriba 'Set default telephoneNo' para que quede clara su función en el contexto de la línea de ensamblaje. A continuación, utilice el botón **Añadir atributo** para crear un atributo con el nombre 'telephoneNo' – el mismo nombre que devolverá el Bucle de conector. Vuelva a pulsar sobre este atributo para establecer el script de asignación:

```
"N/A"
```

Esto significa que a cualquier persona cuyo nombre se lea de la entrada CSV y que no se encuentre en PhoneNumbers.xml se le asignará el valor "N/A" para el atributo 'telephoneNo'²⁶.

Finalmente, incluya este nuevo atributo 'telephoneNo' en la correlación de salida de 'Write_XML_File' arrastrándolo allí y asegurándose de que tenga la asignación:

```
work.telephoneNo
```

La línea de ensamblaje debería ahora tener este aspecto:

26. Si prefiere que estos usuarios no tengan el atributo 'telephoneNo', utilice una asignación que no devuelva ningún valor. Esto se hace devolviendo el valor especial null:

```
null
```

Esto hará que el comportamiento predeterminado ante un valor nulo elimine el atributo de la entrada de trabajo. Como resultado, no llegará a la correlación de salida del conector 'Write_XML_File' y, por lo tanto, no aparecerá en el documento XML resultante.

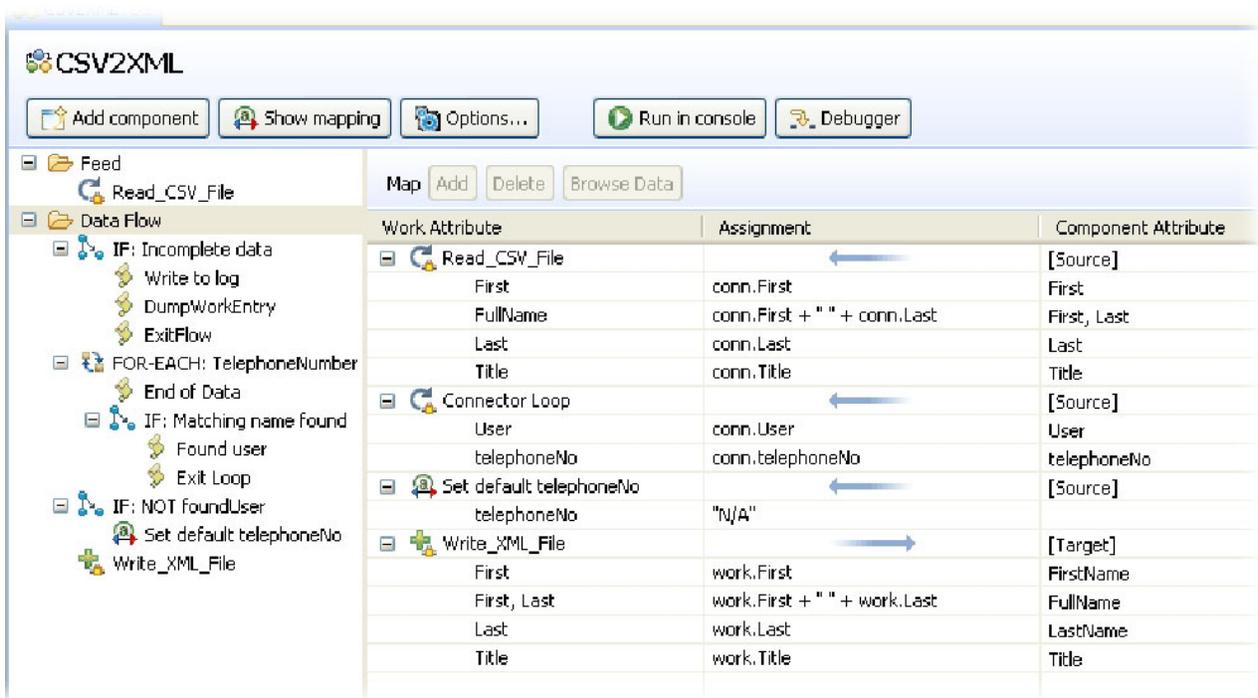
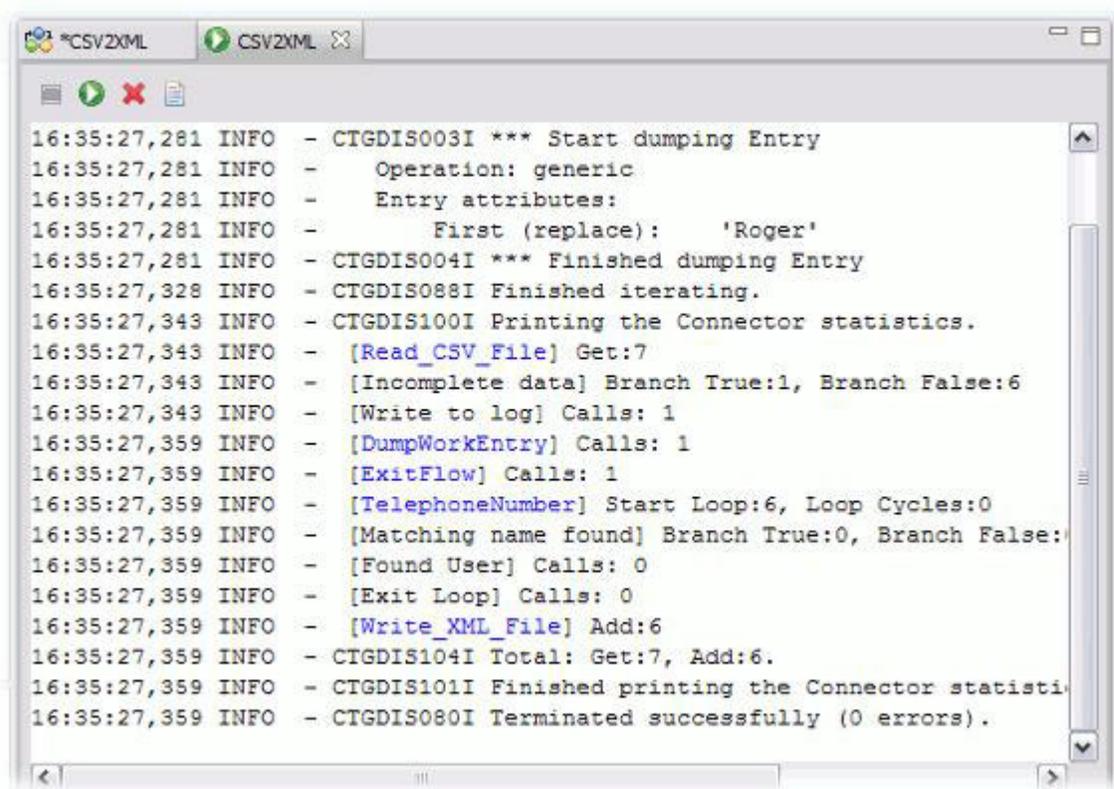


Figura 62. Línea de ensamblaje completa con el bucle FOR-EACH

A continuación, vuelva a ejecutar la LE y examine la salida del registro. La rama 'NOT foundUser' deberá haber dado *verdadero* dos veces y *falso* en las otras cuatro entradas.



```
16:35:27,281 INFO - CTGDIS003I *** Start dumping Entry
16:35:27,281 INFO - Operation: generic
16:35:27,281 INFO - Entry attributes:
16:35:27,281 INFO - First (replace): 'Roger'
16:35:27,281 INFO - CTGDIS004I *** Finished dumping Entry
16:35:27,328 INFO - CTGDIS088I Finished iterating.
16:35:27,343 INFO - CTGDIS100I Printing the Connector statistics.
16:35:27,343 INFO - [Read_CSV_File] Get:7
16:35:27,343 INFO - [Incomplete data] Branch True:1, Branch False:6
16:35:27,343 INFO - [Write to log] Calls: 1
16:35:27,359 INFO - [DumpWorkEntry] Calls: 1
16:35:27,359 INFO - [ExitFlow] Calls: 1
16:35:27,359 INFO - [TelephoneNumber] Start Loop:6, Loop Cycles:0
16:35:27,359 INFO - [Matching name found] Branch True:0, Branch False:0
16:35:27,359 INFO - [Found User] Calls: 0
16:35:27,359 INFO - [Exit Loop] Calls: 0
16:35:27,359 INFO - [Write_XML_File] Add:6
16:35:27,359 INFO - CTGDIS104I Total: Get:7, Add:6.
16:35:27,359 INFO - CTGDIS101I Finished printing the Connector statistics.
16:35:27,359 INFO - CTGDIS080I Terminated successfully (0 errors).
```

Figura 63. Salida del registro con las estadísticas de la rama IF

Observe que algunos nombres de componentes están resaltados (en azul) en las estadísticas de la LE de la salida del registro. Si pulsa sobre estos nombres con el botón izquierdo del ratón mientras mantiene pulsada la tecla Ctrl, se abrirá el componente seleccionado en el editor de líneas de ensamblaje.

La salida XML resultante deberá ser como la siguiente:

```
-<DocRoot>
- <Entry>
  <FirstName>Bill</FirstName>
  <Title>Chief Scientist</Title>
  <LastName>Sanderman</LastName>
  <telephoneNo>N/A</telephoneNo>
  <FullName>Bill Sanderman</FullName>
</Entry>
- <Entry>
  <FirstName>Mick</FirstName>
  <Title>CEO</Title>
  <LastName>Kamerun</LastName>
  <telephoneNo>555-666-6666</telephoneNo>
  <FullName>Mick Kamerun</FullName>
</Entry>
- <Entry>
  <FirstName>Jill</FirstName>
  <Title>CTO</Title>
```

Figura 64. Salida XML con el atributo 'telephoneNo'

Hasta ahora, todo bien. Ha llegado el momento de probar a utilizar la modalidad *Buscar* para llevar a cabo la unión.

Utilización de la modalidad **Buscar**

En vez de modificar directamente la línea de ensamblaje que tiene en ejecución, con el consiguiente peligro de estropearla, haremos una copia de la LE y modificaremos la copia. Para hacerlo, pulse el botón derecho del ratón sobre la línea de ensamblaje 'CSV2XML' y seleccione **Copiar**.

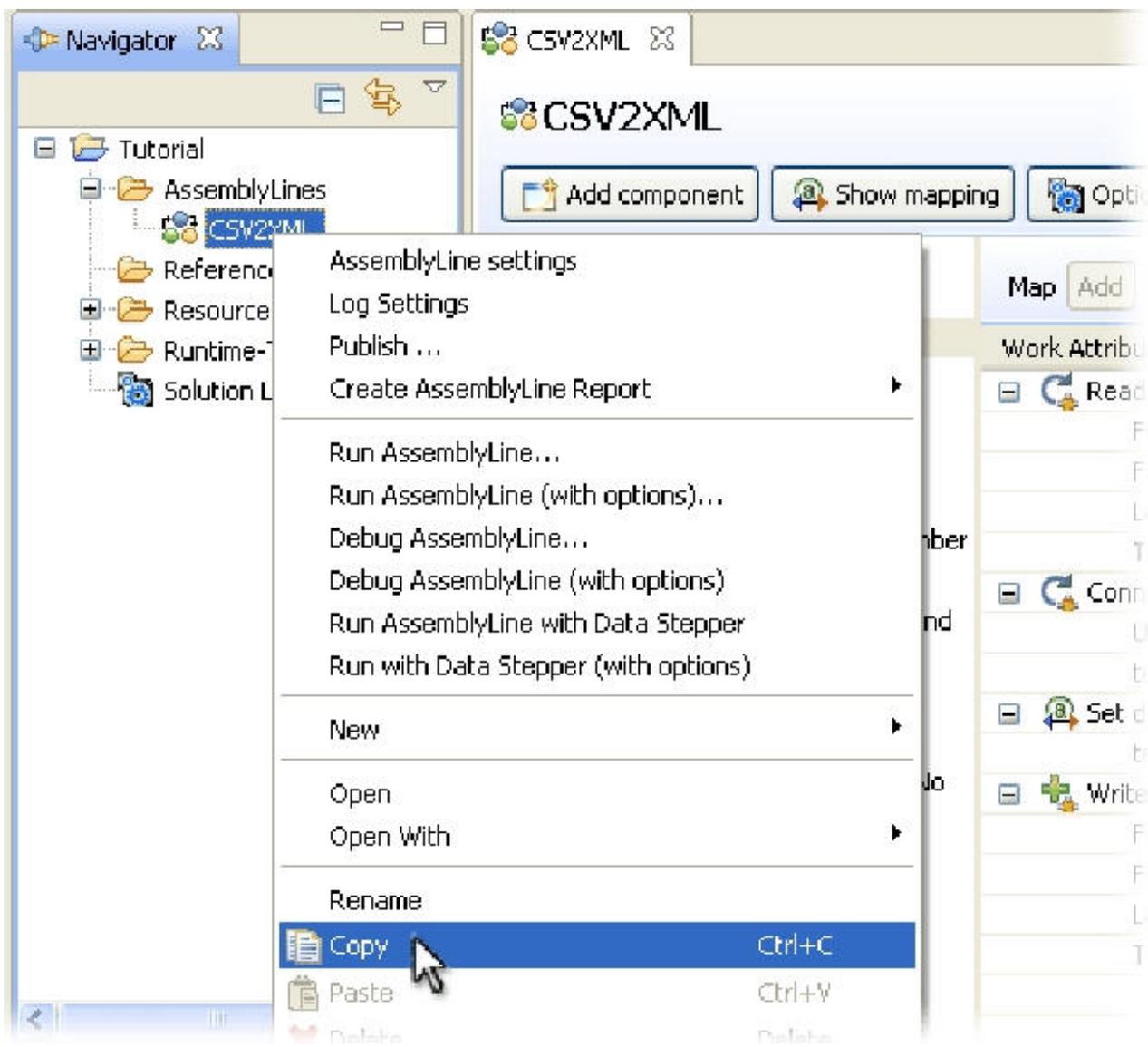


Figura 65. Función para copiar la línea de ensamblaje

A continuación, pulse con el botón derecho del ratón sobre la carpeta 'Líneas de ensamblaje' en el Navegador y seleccione la opción **Pegar**. Cambie el nombre de esta nueva línea de ensamblaje a 'CSV2XML_LookupMode' y efectúe una doble pulsación sobre la misma para abrir el editor de LE.

A continuación, elimine el bucle de conector junto con todos los componentes que están bajo el mismo. Para hacerlo, selecciónelo y pulse la tecla **Suprimir**. En su lugar, arrastraremos un conector JDBC que copiaremos de otra línea de ensamblaje.

Pero antes debe crear la tabla de base de datos de la que este conector leerá los datos; o mejor, ejecutar una línea de ensamblaje preconstruída que la creará para usted. Para hacerlo, utilice un navegador de archivos para navegar hasta la carpeta 'Tutorials' y busque el archivo CreatePhoneDB. Arrástrelo a la ventana del CE, sobre la carpeta 'Líneas de ensamblaje' en el panel del navegador.

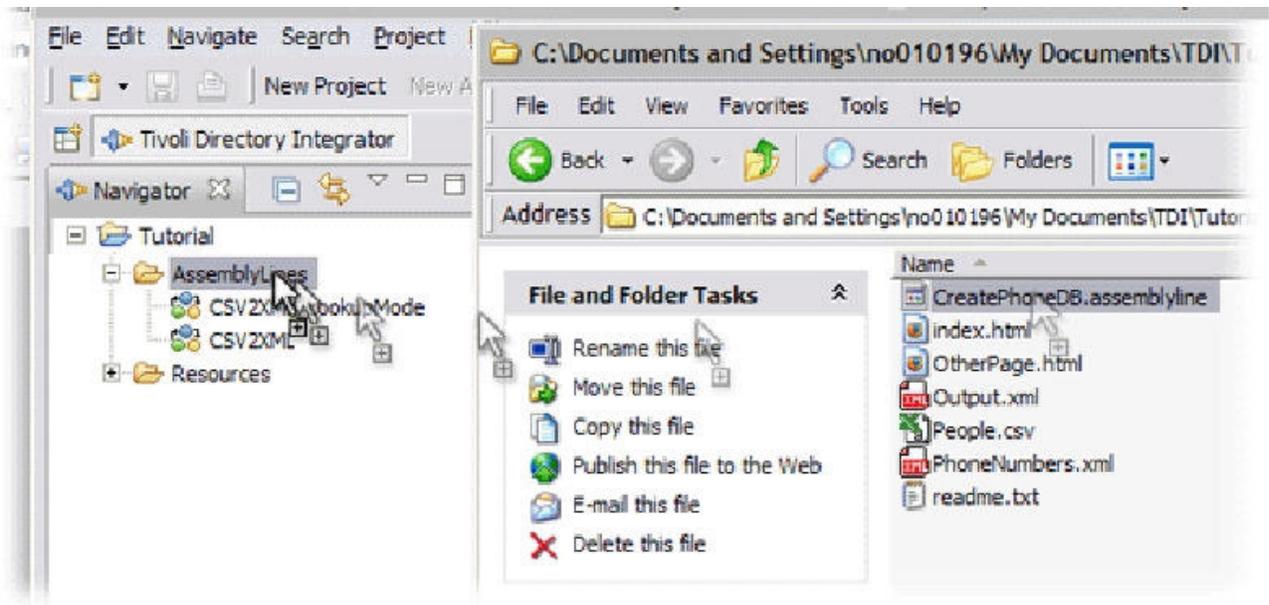


Figura 66. Copiar una línea de ensamblaje al proyecto

La línea de ensamblaje se importará al proyecto. A continuación, pulse con el botón derecho del ratón sobre la línea de ensamblaje recién importada y seleccione **Ejecutar línea de ensamblaje...**

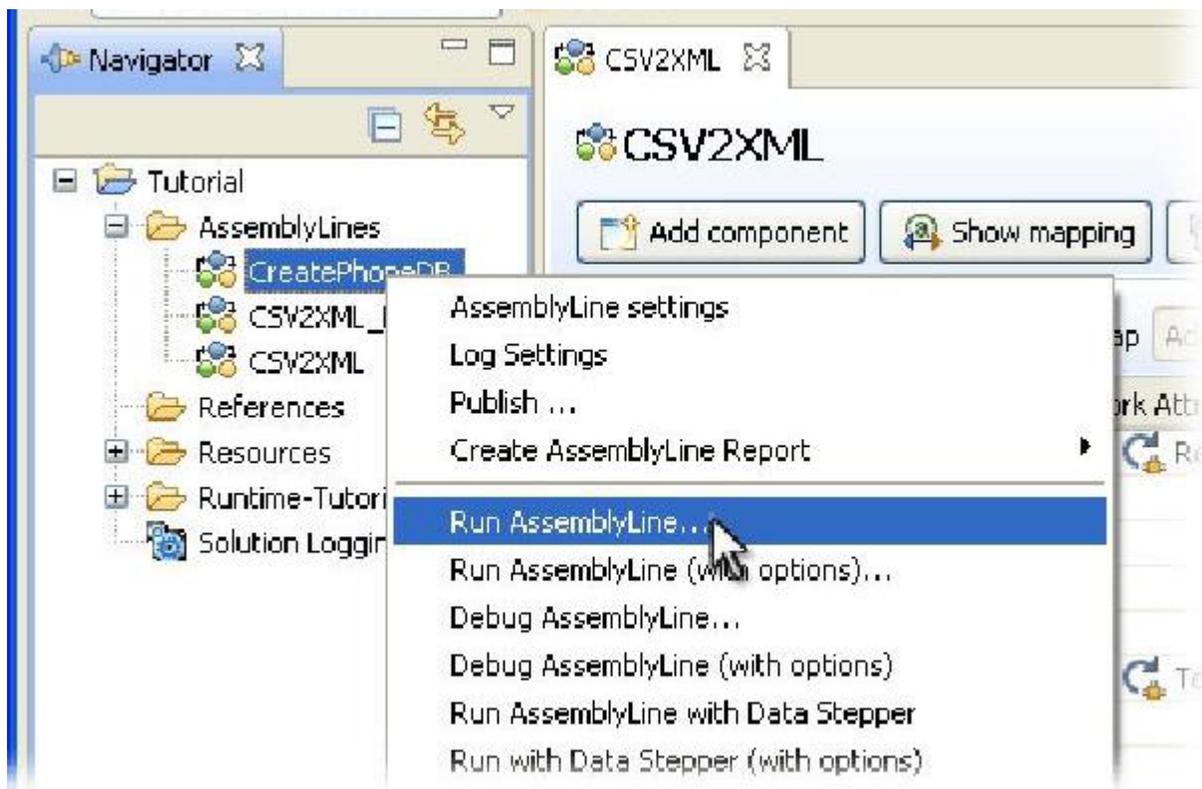
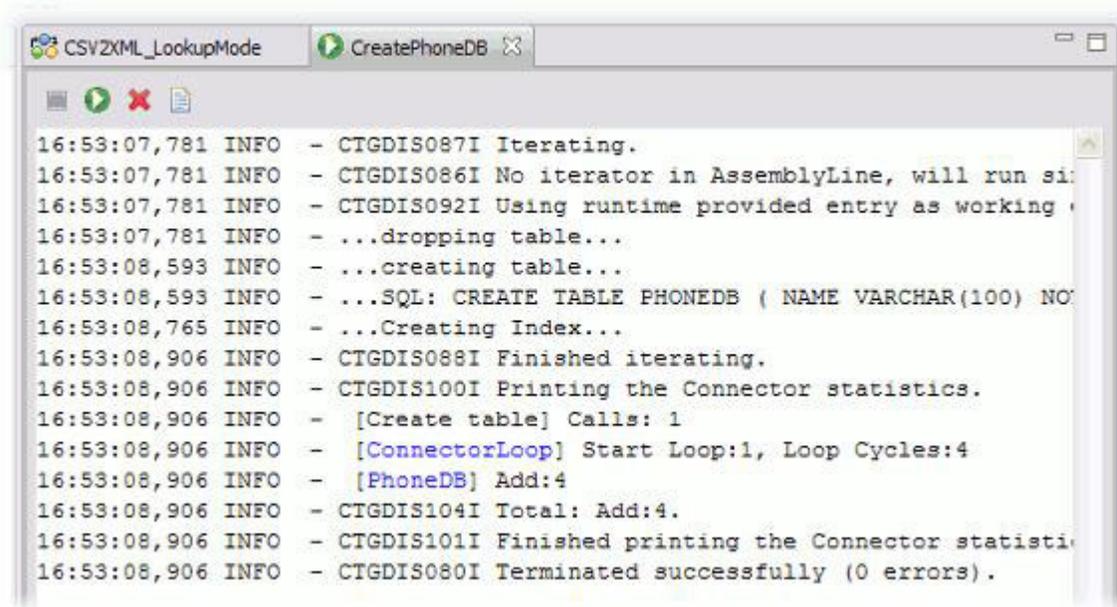


Figura 67. Ejecutar la LE CreatePhoneDB

Esta línea de ensamblaje creará primero una Derby²⁷ en el directorio de soluciones denominado 'TutorialDB' y después creará una tabla 'PhoneDB'. A continuación, recorrerá el archivo PhoneNumbers.xml en bucle y cargará esta información en la nueva tabla²⁸.

Si todo va bien, la salida del registro debería ser como la siguiente:



```
16:53:07,781 INFO - CTGDIS087I Iterating.
16:53:07,781 INFO - CTGDIS086I No iterator in AssemblyLine, will run si
16:53:07,781 INFO - CTGDIS092I Using runtime provided entry as working
16:53:07,781 INFO - ...dropping table...
16:53:08,593 INFO - ...creating table...
16:53:08,593 INFO - ...SQL: CREATE TABLE PHONEDB ( NAME VARCHAR(100) NO
16:53:08,765 INFO - ...Creating Index...
16:53:08,906 INFO - CTGDIS088I Finished iterating.
16:53:08,906 INFO - CTGDIS100I Printing the Connector statistics.
16:53:08,906 INFO - [Create table] Calls: 1
16:53:08,906 INFO - [ConnectorLoop] Start Loop:1, Loop Cycles:4
16:53:08,906 INFO - [PhoneDB] Add:4
16:53:08,906 INFO - CTGDIS104I Total: Add:4.
16:53:08,906 INFO - CTGDIS101I Finished printing the Connector statisti
16:53:08,906 INFO - CTGDIS080I Terminated successfully (0 errors).
```

Figura 68. Salida del registro de la línea de ensamblaje 'CreatePhoneDB'

La línea de ensamblaje 'CreatePhoneDB' tiene un conector JDBC que ya está configurado y a punto para ser utilizado. Deberá copiar este componente a la biblioteca de recursos de su proyecto (concretamente, a la carpeta "Conectores") y reutilizarlo en su LE.

Abra la línea de ensamblaje 'CreatePhoneDB', coja el conector denominado 'PhoneDB' y arrástrelo a la carpeta 'Conectores' que encontrará bajo 'Recursos' en el árbol del navegador.

27. Apache Derby es una base de datos relacional de código abierto que se entrega con IBM Security Directory Integrator.

28. Aunque en esta guía no se tratará esta LE, constituye un buen ejemplo de las técnicas de creación avanzada de scripts utilizadas para explotar la funcionalidad específica de los orígenes de datos que se encuentran en la mayoría de las interfaces de conector. No dude a examinar y probar esta línea de ensamblaje tanto como desee.

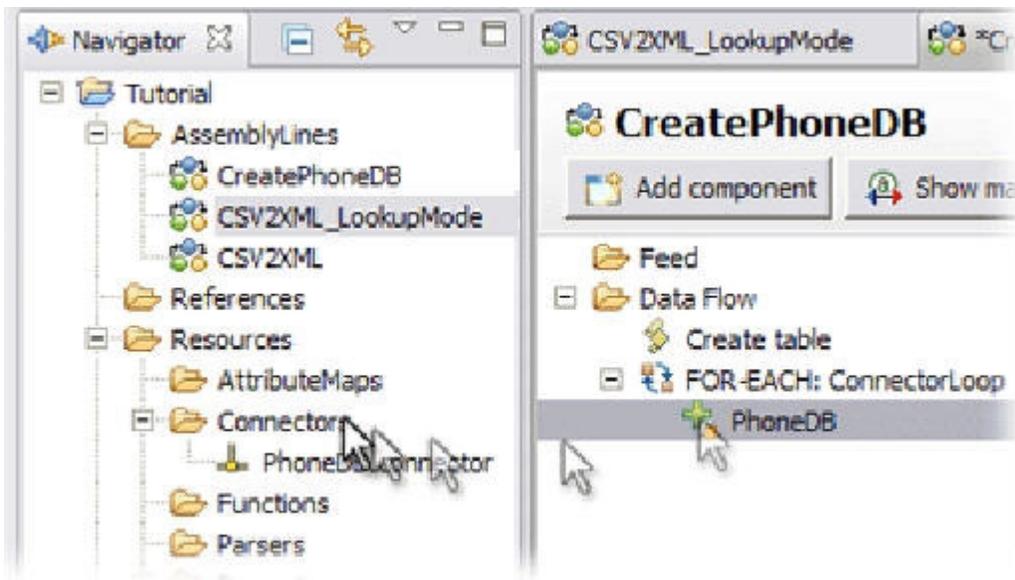


Figura 69. Arrastrar un conector a Recursos

Cierre 'CreatePhoneDB' para ver otra vez la línea de ensamblaje. A continuación, arrastre el nuevo recurso 'PhoneDB' hasta el punto que anteriormente ocupaba el bucle de conector.

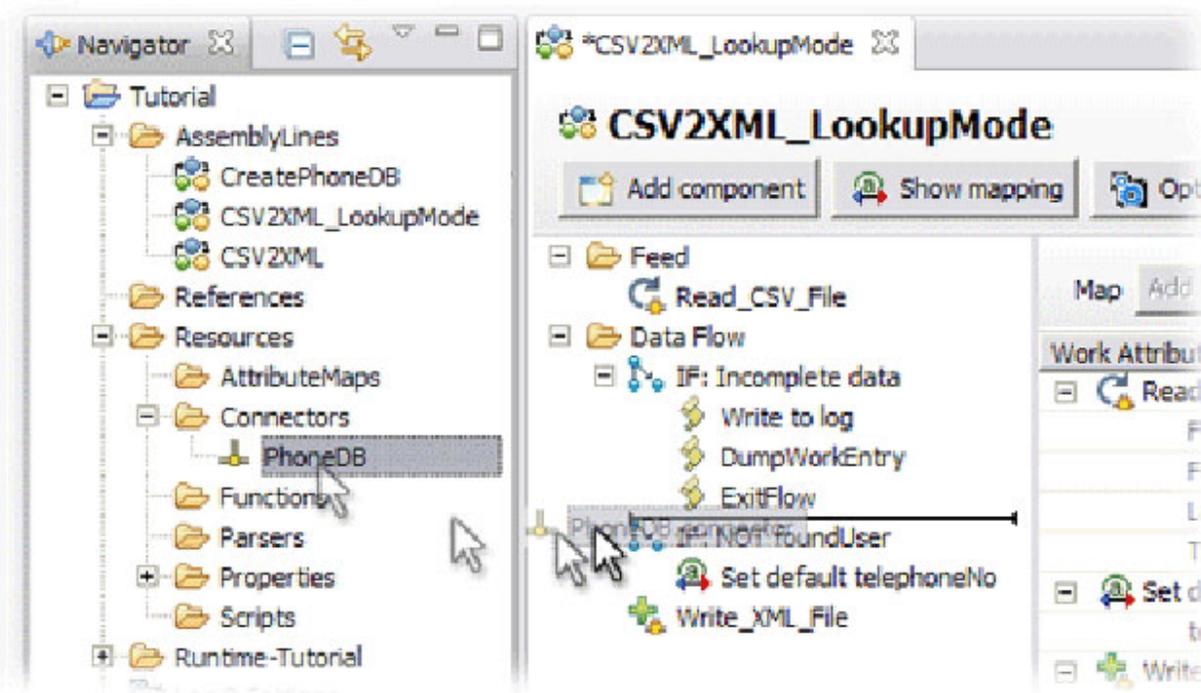


Figura 70. Arrastrar el nuevo recurso a la línea de ensamblaje

Herencia

Habrá observado que este conector aparece de color azul en la línea de ensamblaje. Esto es porque, actualmente, hereda de la biblioteca de recursos. Esto significa que

recuperará dinámicamente los valores de configuración en tiempo de ejecución del conector que ha arrastrado. Esta característica de herencia facilita la reutilización de recursos, como, por ejemplo, componentes configurados y lógica en script, entre distintas líneas de ensamblaje.

Puede cambiar el ancestro de un componente con el botón **Heredar de** en la parte superior de este panel de editor. Las pestañas de componente, como Conexión, Delta, Correlaciones de entrada y salida y Enganches también proporcionan una opción para definir la herencia, permitiéndole definir un ancestro distinto al del propio componente.

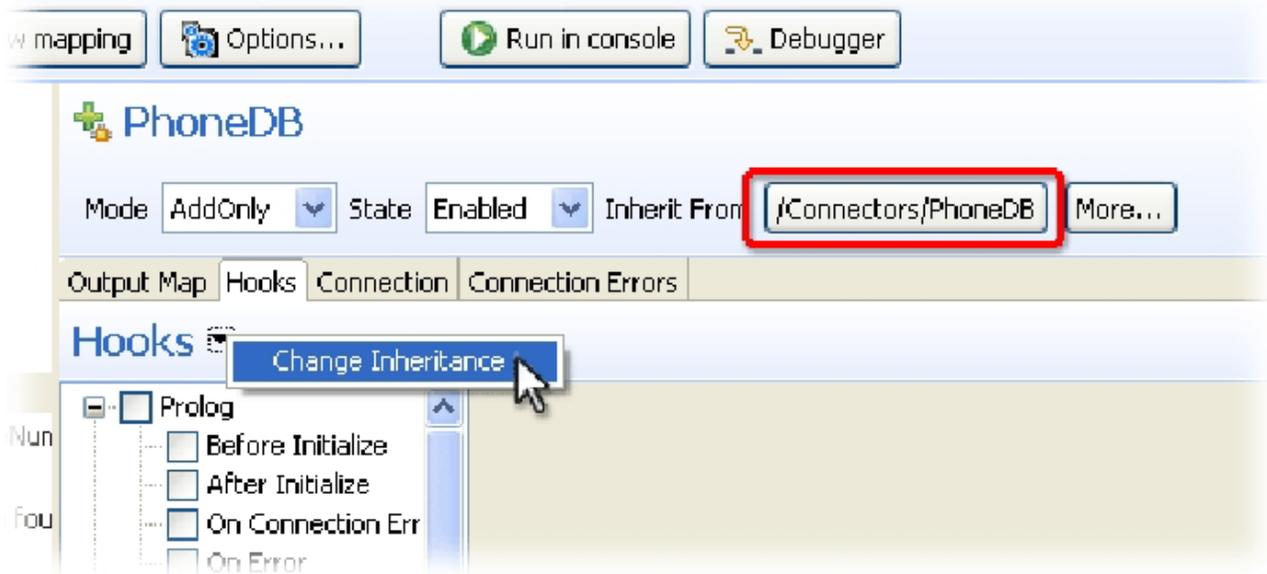


Figura 71. Establecer la herencia para la pestaña enganches

Los valores heredados se muestran en azul. Si los cambia, se pierde la herencia. La herencia se puede restaurar utilizando la opción **Revertir al valor heredado** en los menús contextuales de las reglas de correlación de atributos y de los enganches.

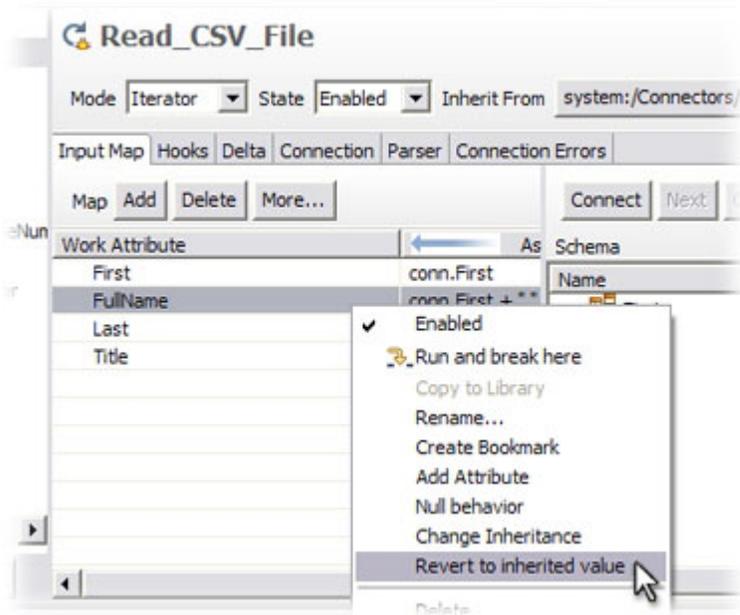


Figura 72. Restaurar la herencia en una regla de correlación

Volviendo otra vez al ejercicio, cambie primero el valor **Modalidad** del nuevo conector "PhoneDB" de **Sólo adición** a **Buscar**.

En segundo lugar, puesto que la correlación de atributos originalmente era una *Correlación de salida* asociada a la modalidad anterior, tendrá que descubrir el esquema de entrada pulsando los botones **Conectar** y **Siguiente** en el Esquema de conector. El tercer y último paso consistirá en arrastrar el atributo 'PHONE' desde el Esquema hasta la Correlación de entrada, con lo que obtendrá una correlación simple para este valor.

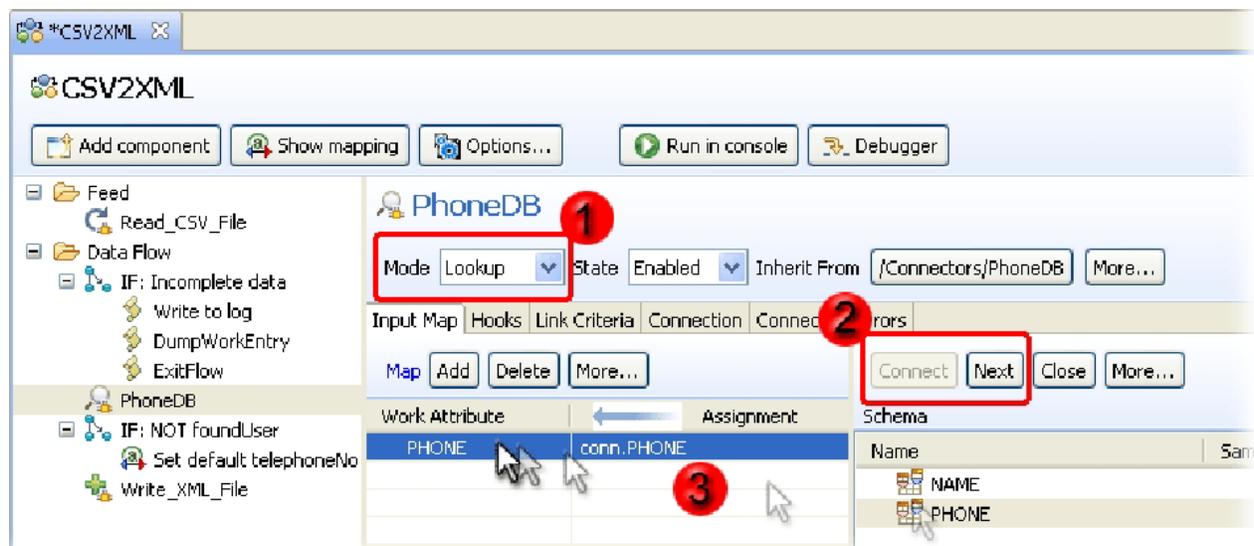


Figura 73. Cambiar la modalidad, descubrir y correlacionar atributos

Una vez tenga la regla de correlación de entrada, es importante que le cambie el nombre de 'PHONE' a 'telephoneNo' para que se adecue a la asignación de la correlación de salida del conector 'Write_XML_File'.

Por si se lo ha preguntado, no es necesario que arrastre también 'NAME' a la correlación de entrada, puesto que ya tiene el nombre del usuario. Este campo sí se utilizará, sin embargo, al definir la regla de búsqueda.

Reglas de búsqueda = Criterios de enlace

Cuando ha seleccionado la modalidad *Búsqueda* ha aparecido una nueva pestaña 'Criterio de enlace' en el editor del conector. Los criterios de enlace sirven para definir las reglas de coincidencia para la búsqueda.

Puede definirlos como criterios de enlace simples utilizando los desplegable, añadiendo los criterios de enlace que necesite y seleccionando el recuadro **Coincidir con cualquiera** como haría para las condiciones. También puede seleccionar el recuadro de selección **Crear criterio con script personalizado** y escribir un fragmento de código script que calcule la regla de búsqueda, como en este ejemplo de un filtro de búsqueda LDAP:

```
"(cn=" + work.FullName + ")"
```

Observe que esta solución hará que su solución y el origen de datos donde se realice la búsqueda estén más fuertemente vinculados puesto que deberá escribir la sintaxis que espere el sistema conectado. En nuestro caso, esto equivaldría a crear una cláusula WHERE (sin la palabra clave 'WHERE').

En cambio, en el caso de los criterios de enlace simples, el conector los traduce a la sintaxis de búsqueda nativa, de forma que puede cambiar la interfaz de conector sin tener que rehacer los criterios de enlace.

Los criterios de enlace simples tienen un aspecto parecido a las condiciones. El primer desplegable se llena con el esquema que ha descubierto y el segundo le muestra los atributos de la entrada de trabajo que están disponibles en este momento en la LE. Aquí también, lo mismo que en el caso de las condiciones, el signo dólar se utiliza para indicar que el valor del atributo mencionado se debe sustituir en tiempo de ejecución para crear el filtro de búsqueda.



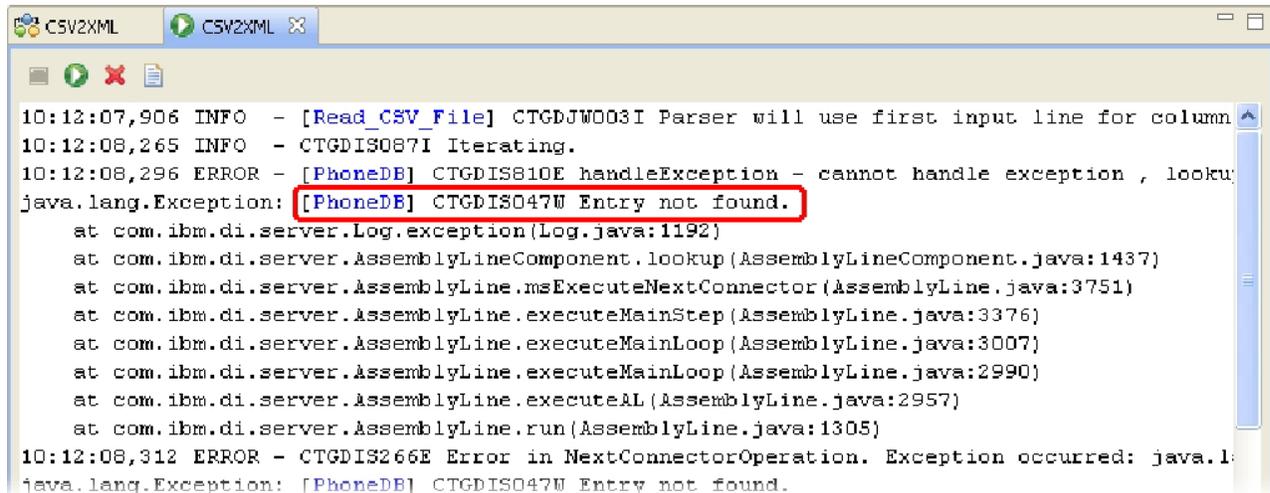
Figura 74. Un criterio de enlace simple

Recuerde guardar su trabajo de vez en cuando pulsando CTRL-S²⁹. Debe hacerlo regularmente para no perder su trabajo.

Es el momento de volver a Ejecutar la línea de ensamblaje.

Descifrar errores de ejecución

No se asuste. Habrá obtenido un error, que se indica por el volcado de pila que aparece en la salida del registro. Desplácese hasta el principio del primer volcado de pila. Encontrará información sobre *dónde* ha ocurrido el problema así como sobre *qué* lo ha provocado.



```
10:12:07,906 INFO - [Read_CSV_File] CTGDJW003I Parser will use first input line for column
10:12:08,265 INFO - CTGDIS087I Iterating.
10:12:08,296 ERROR - [PhoneDB] CTGDIS810E handleException - cannot handle exception , lookup
java.lang.Exception: [PhoneDB] CTGDIS047W Entry not found.
    at com.ibm.di.server.Log.exception(Log.java:1192)
    at com.ibm.di.server.AssemblyLineComponent.lookup(AssemblyLineComponent.java:1437)
    at com.ibm.di.server.AssemblyLine.msExecuteNextConnector(AssemblyLine.java:3751)
    at com.ibm.di.server.AssemblyLine.executeMainStep(AssemblyLine.java:3376)
    at com.ibm.di.server.AssemblyLine.executeMainLoop(AssemblyLine.java:3007)
    at com.ibm.di.server.AssemblyLine.executeMainLoop(AssemblyLine.java:2990)
    at com.ibm.di.server.AssemblyLine.executeAL(AssemblyLine.java:2957)
    at com.ibm.di.server.AssemblyLine.run(AssemblyLine.java:1305)
10:12:08,312 ERROR - CTGDIS266E Error in NextConnectorOperation. Exception occurred: java.l
java.lang.Exception: [PhoneDB] CTGDIS047W Entry not found.
```

Figura 75. Mensaje de error en la salida del registro

El nombre del componente aparece entre corchetes ('PhoneDB') seguido de la descripción del error indicando que no se ha encontrado la entrada – en otras palabras, que la búsqueda ha fallado.

Cuando se configura un conector en modalidad *Buscar*, el sistema espera encontrar un registro coincidente (uno y sólo uno) al realizar la búsqueda. Si no se encuentra ninguno - o si hay diversos registros que coinciden con el criterio de enlace - va a parar a los *Enganches* especiales, que deben estar por lo menos habilitados para evitar que la LE se detenga. Este comportamiento es claramente visible en los diagramas de Flujo de datos que aparecen en *Referencia*. A continuación se muestra un extracto de la página donde se explica la modalidad *Buscar*:

29. Si ha suprimido líneas de ensamblaje o recursos y desea deshacer esta acción, pulse con el botón derecho del ratón sobre el proyecto en el panel del Navegador y seleccione **Restaurar del historial local...** y se le mostrará una lista de las versiones de activos desde donde puede restaurar. Por supuesto, para que algo aparezca en su historial local, debe haberlo guardado anteriormente.

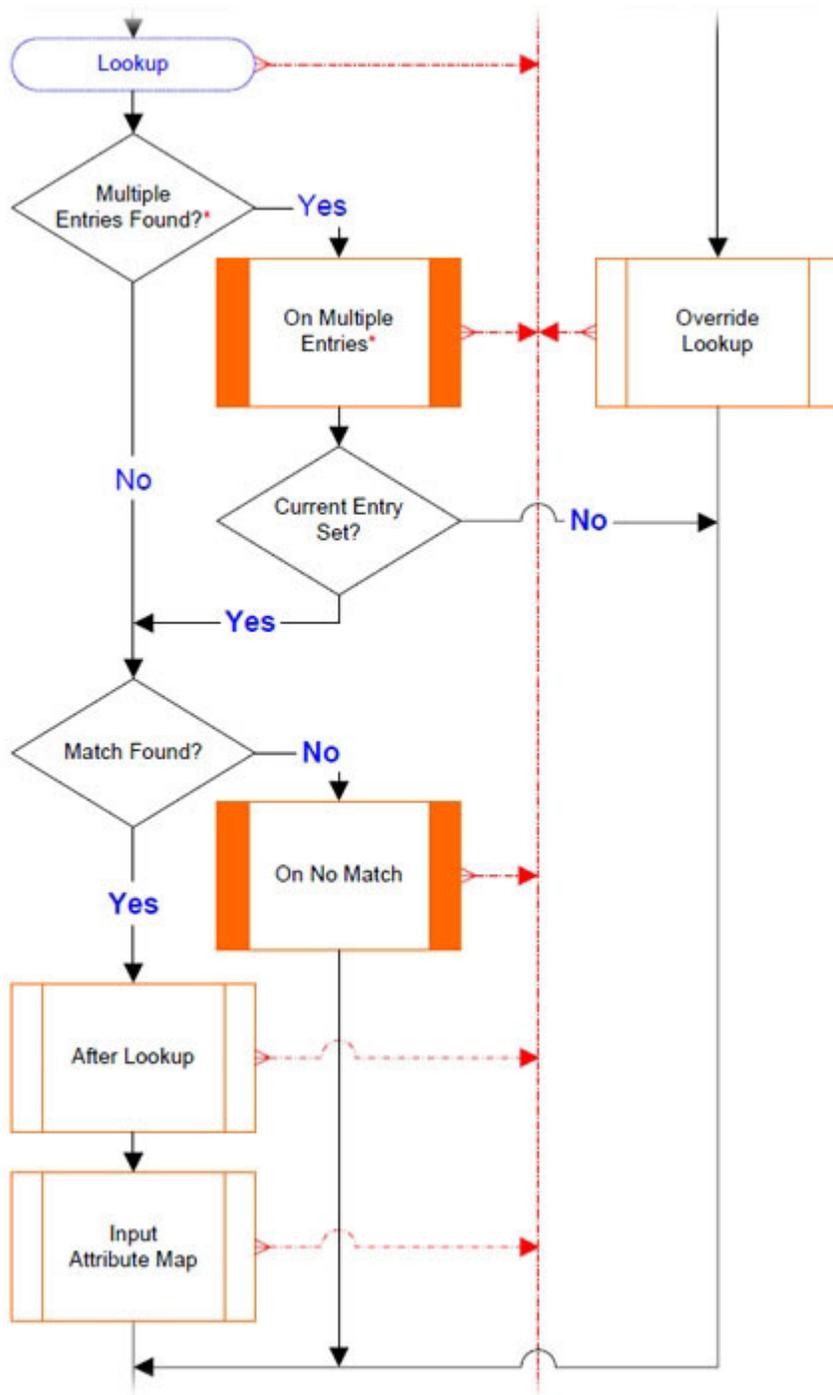


Figura 76. Diagrama de flujo parcial de la modalidad Buscar

Puede aprovechar este comportamiento para establecer la variable del distintivo foundUser. Pulse con el botón derecho del ratón sobre el conector 'PhoneDB' y seleccione **Enganches...** para abrir el editor de enganches. Seleccione el enganche 'En caso de ninguna coincidencia' y escriba el código script para establecer foundUser a *false*.

```
foundUser = false;
```

A continuación, seleccione 'Después de Lookup' y escriba el siguiente script de enganche complementario³⁰:

```
foundUser = true;
```

La línea de ensamblaje debería ahora parecerse a la de la imagen:

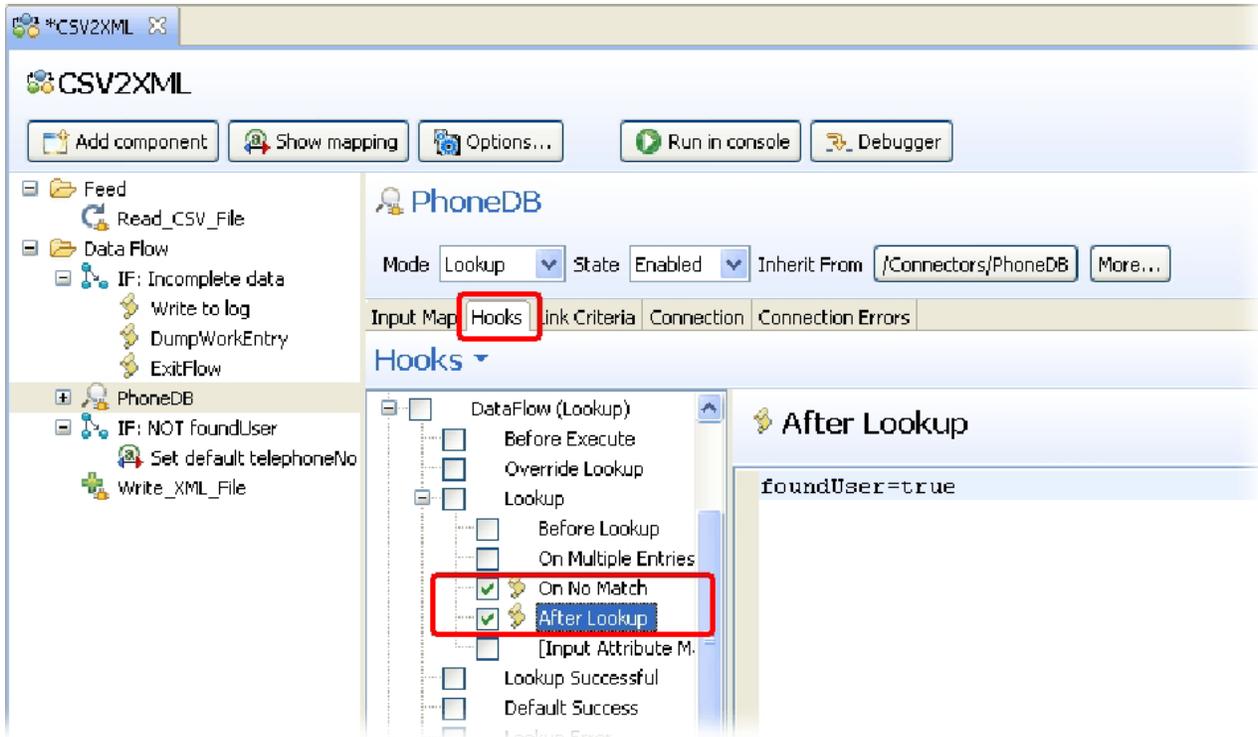


Figura 77. Primer ejercicio de la guía de aprendizaje completado

Ha llegado el momento de volver a **Ejecutar** la línea de ensamblaje, y esta vez debería terminar sin errores. El archivo `Output.xml` debería ser idéntico al obtenido en la LE basada en bucle de la sección anterior.

¡Felicidades! Ha terminado el primer ejercicio de la guía de aprendizaje de IBM Security Directory Integrator. Ha llegado el momento de ver cómo desencadenar las líneas de ensamblaje a partir de sucesos en tiempo real.

30. Como puede ver en el diagrama de flujo, el enganche 'Después de Lookup' sólo se ejecuta si la búsqueda da como resultado una única coincidencia.

Capítulo 3. Integración dirigida por sucesos

Hasta ahora ha estado ejecutando las líneas de ensamblaje como procesos por lotes, iniciándolos manualmente cada vez que deseaba que fluyeran los datos. Puede también ejecutar líneas de ensamblaje desde la línea de mandatos invocando al servidor de IBM Security Directory Integrator, de esta manera³¹:

```
ibmdisrv -c examples/Tutorial/Tutorial1.xml -r CSVtoXML
```

De esta forma, resulta fácil utilizar herramientas de planificación (como, por ejemplo, *crontab*) para planificar su funcionamiento o para lanzarlas fácilmente desde aplicaciones externas.

IBM Security Directory Integrator proporciona diversas características para hacer que sus líneas de ensamblaje reconozcan sucesos, permitiendo que sus soluciones manejen y respondan a una amplia variedad de desencadenantes en tiempo real.

Ejemplos de estos desencadenantes son:

- solicitudes de protocolo procedentes de un puerto IP, como llamadas REST, SNMP y servicios web;
- nuevos mensajes que aparezcan en una cola;
- mensajes de correo electrónico que lleguen a una bandeja de entrada;
- cambios en los datos, por ejemplo en archivos, bases de datos, directorios y bases de datos de Notes;
- operaciones de línea de ensamblaje basadas en planificaciones o en temporizadores.

Esta no es una lista completa, y encontrará inspiración y orientación en otras publicaciones de IBM Security Directory Integrator, en los sitios web de comunidad y en los grupos de noticias.

La gestión de estos sucesos en su solución se puede realizar de diversas formas:

Conectores en modalidad Iterador

Algunos conectores permiten configurar parámetros de tiempo de espera para la modalidad Iterador. Un ejemplo es el conector del sistema de archivos, que puede configurarse para leer en un archivo hasta el final y posteriormente esperar a que aparezca la información nueva - denominada 'lectura final'.

Otros conectores, como los correspondientes a la detección de cambios RDBMS y al registro de cambios de LDAP, funcionan de forma parecida. Estos conectores permiten crear líneas de ensamblaje que se ejecutan de forma continua, esperando a que aparezcan nuevos cambios en el sistema conectado.

También hay un conector de temporizador que se ejecuta en modalidad Iterador y se puede configurar para dirigir su línea de ensamblaje a intervalos regulares de acuerdo con un parámetro de planificación. Probará este último en breve.

31. El servidor de IBM Security Directory Integrator proporciona un mensaje de sintaxis cuando se invoca sin argumentos de línea de mandatos: `ibmdisrv`

IBM Security Directory Integrator incluye una Consola de administración web como parte de la instalación estándar. Esta aplicación basada en navegador, denominada la Consola de administración y supervisión (AMC) le permite supervisar el estado de las líneas de ensamblaje, cargar configuraciones dinámicamente en servidores en ejecución, iniciar y detener LE y configurar el comportamiento ante error o de respuesta para mantener sus soluciones de integración altamente disponibles. También se puede utilizar para planificar cuándo se deben ejecutar las líneas de ensamblaje. Sin embargo, la herramienta de administración web no forma parte del contenido de esta guía.

Nota: La característica AMC está en desuso y se eliminará en una versión futura de IBM Security Directory Integrator.

Conectores en modalidad de servidor

Algunos conectores especializados, como el conector de servidor HTTP y el conector de servidor LDAP, le permiten crear soluciones que procesen solicitudes de entrada procedentes de clientes externos, realicen las acciones solicitadas y respondan con las respuestas adecuadas. En el siguiente ejercicio, utilizará el conector de servidor HTTP.

Notificaciones y propiedades

IBM Security Directory Integrator tiene componentes que pueden suscribirse a sucesos de notificación de IBM Security Directory Integrator, del mismo modo que existen componentes (y llamadas de script) para enviar estos sucesos – incluso entre distintos servidores de IBM Security Directory Integrator que se ejecuten en plataformas distintas.

En esta guía se explica de forma detallada la planificación de LE y la modalidad de servidor para conectores.

Planificar líneas de ensamblaje

Usando la interfaz de usuario en el editor de configuraciones, puede crear un planificador para ejecutar una línea de ensamblaje en momentos específicos. Por ejemplo, puede crear un planificador que se ejecute todos los días a las 3:05 a.m., todos los sábados a las 7:00 a.m., o incluso para planificaciones más complejas.

Para crear un planificador, en el editor de configuraciones, pulse **Archivo > Nuevo > Planificador**. Como alternativa, pulse con el botón derecho en AssemblyLine y seleccione **Crear planificación**.

Cuando el servidor IBM Security Directory Integrator carga un archivo de configuración, se inician automáticamente los planificadores habilitados que están contenidos en el archivo de configuración. El mandato para cargar el archivo de configuración es `ibmdisrv -c myconfig.xml -d`, donde `myconfig.xml` es el archivo de configuración exportada. Si detiene la instancia de configuración, se detienen todos los planificadores asociados.

Figura 78. Planificador de IBM Security Directory Integrator

Los detalles sobre el planificador de IBM Security Directory Integrator se encuentran en la sección *Planificador de IBM Security Directory Integrator de Referencia*.

Líneas de ensamblaje de solicitud de servicio

En este último ejercicio, deberá crear una línea de ensamblaje de servidor web para proporcionar una interfaz de usuario muy simple para iniciar la LE 'CSV2XML_LookupMode'. En otras palabras, un servicio HTTP para iniciar transferencias de datos.

Para empezar, cree una nueva LE con el nombre 'TINA_WebServer'³². A continuación, añada un nuevo componente, seleccionando el 'Conector de servidor HTTP' y pulse **Finalizar** para salir del asistente. A continuación, abra la pestaña

32. TINA aquí significa "This Is Not Apache" :-)

Correlación de entrada.

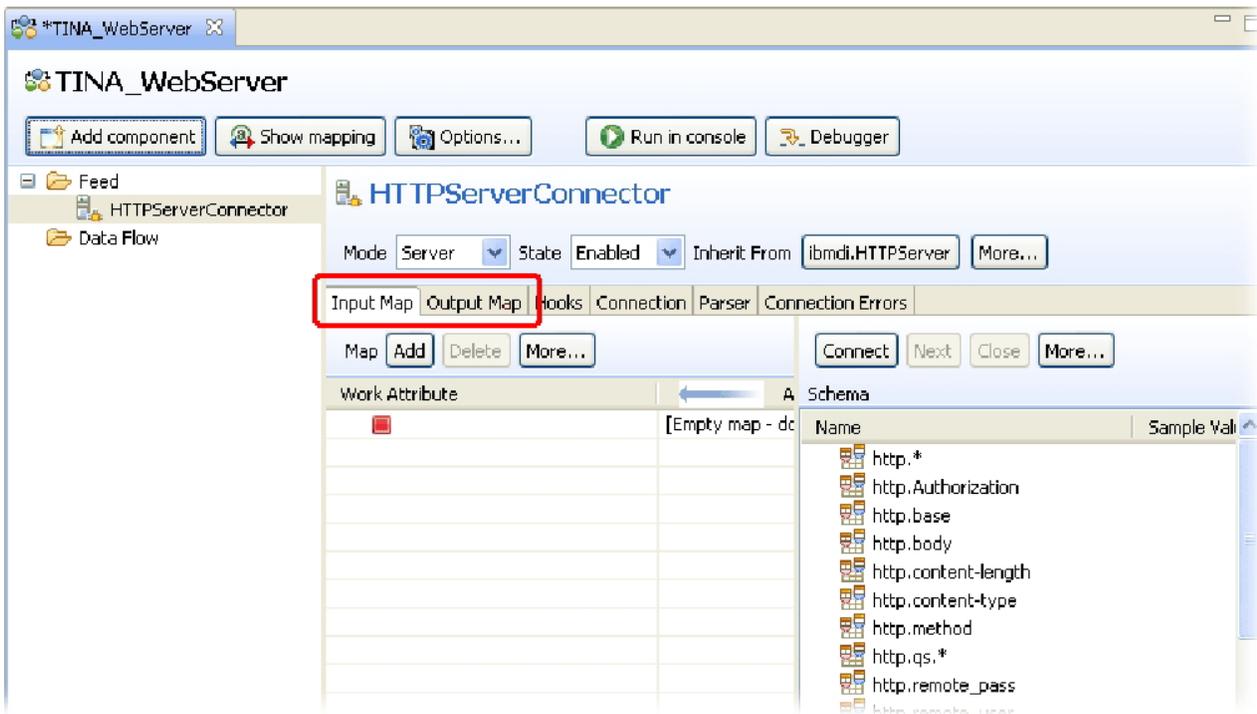


Figura 79. Panel de correlación de atributos del Conector de servidor HTTP

Los conectores de servidor son complementarios a los componentes de función. Mientras los componentes de función realizan solicitudes de servicio, un conector de servidor proporciona y alimenta un servicio. En consecuencia, la correlación de entrada para un conector de servidor se utiliza para recibir atributos procedentes de un cliente que realiza una solicitud, mientras que la correlación de salida proporciona una forma para responder. También puede ver, en la parte derecha de la pantalla de correlación de atributos que el esquema de entrada ya está creado. Lo mismo ocurre con el esquema de salida. Los conectores de servidor proporcionan esta información para ayudarle a realizar la correlación. Observe, sin embargo, que algunos de estos atributos de esquema contienen caracteres comodín, como 'http.qs.*'. Se trata de información del momento del diseño que le indica que puede esperar cualquier número de atributos entrantes cuyos nombres empiecen por 'http.qs'.³³

Finalmente, el desplegable **Modalidad** para todos los conectores de servidor ofrece la modalidad de servidor y la modalidad de iterador. Hay una modalidad adicional (Respuesta) que no se muestra aquí, con lo que suman un total de tres posibles modalidades. El conector de servidor cambia de una modalidad a otra en diversos momentos durante su funcionamiento.

1. Un conector de servidor empieza primero en modalidad *Servidor*, conectándose a algún recurso, como un puerto IP y esperando a que haya conexiones de cliente entrantes;
2. Una vez se ha establecido una conexión, el conector pasa a modalidad *Iterador* para recuperar datos del cliente en base a la correlación de entrada y pasarlos a los componentes del flujo de datos;

33. Este conjunto de atributos en particular (http.qs.*) transportarán cualesquiera parámetros de serie de consulta que el cliente pase a la llamada HTTP.

- Finalmente, cuando los componentes del *Flujo de datos* se han terminado de ejecutar, el conector pasa a modalidad respuesta, utilizando la correlación de salida para conformar una respuesta y enviarla de vuelta al cliente.

No tiene que preocuparse por esto, ya que se maneja automáticamente. Sin embargo, si hace scripts de enganches, verá que hay tres tipos de enganches: *Servidor*, *Iterador* y *Respuesta*.

Continúe el ejercicio de aprendizaje añadiendo un elemento atributo en la correlación de entrada.

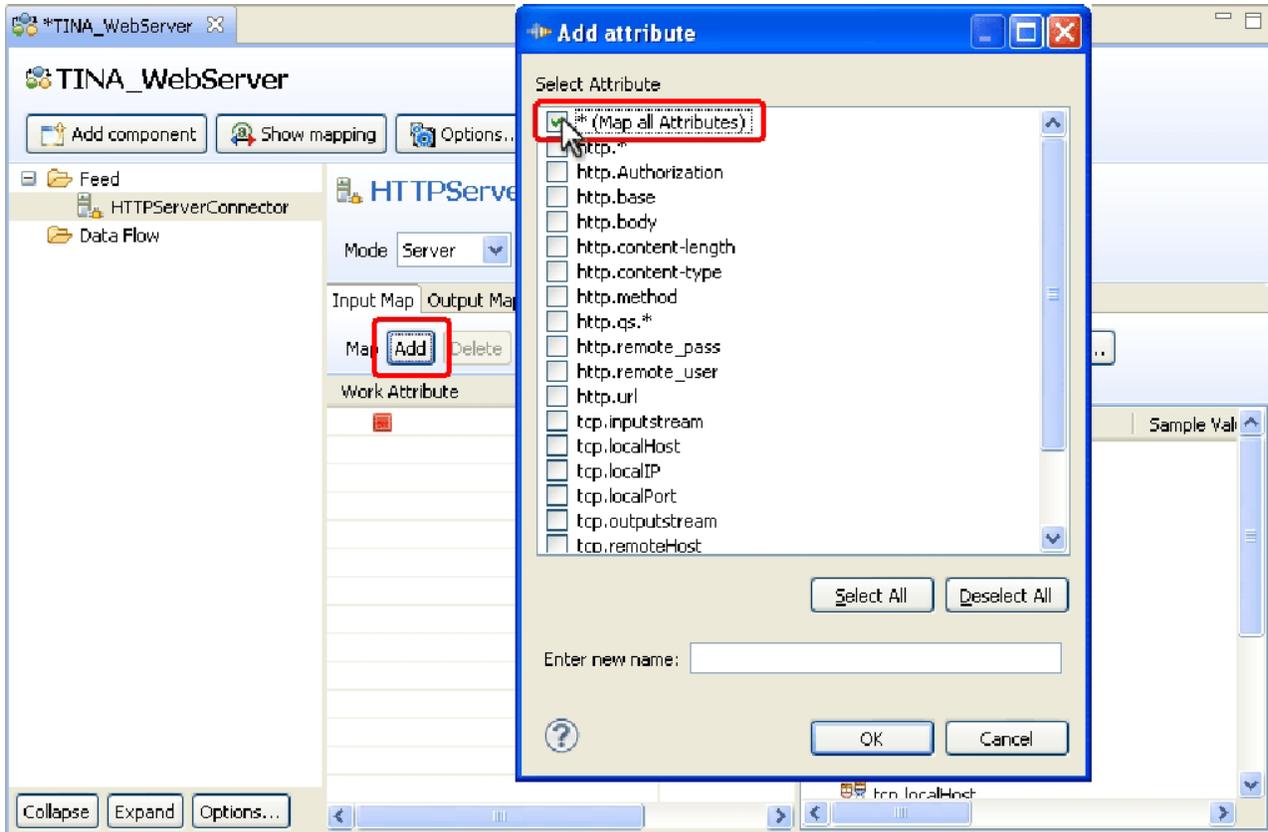


Figura 80. Añadir un elemento de correlación atributos de entrada

Simplemente seleccione la opción situada en la parte superior del diálogo presentado. En lugar de esto, también puede introducir el carácter de asterisco (*) en el campo Escribir nuevo nombre. Esta es la regla especial de correlación comodín que indica a IBM Security Directory Integrator que correlacione todos los atributos leídos por el conector.

Su correlación de entrada tendrá el aspecto siguiente:

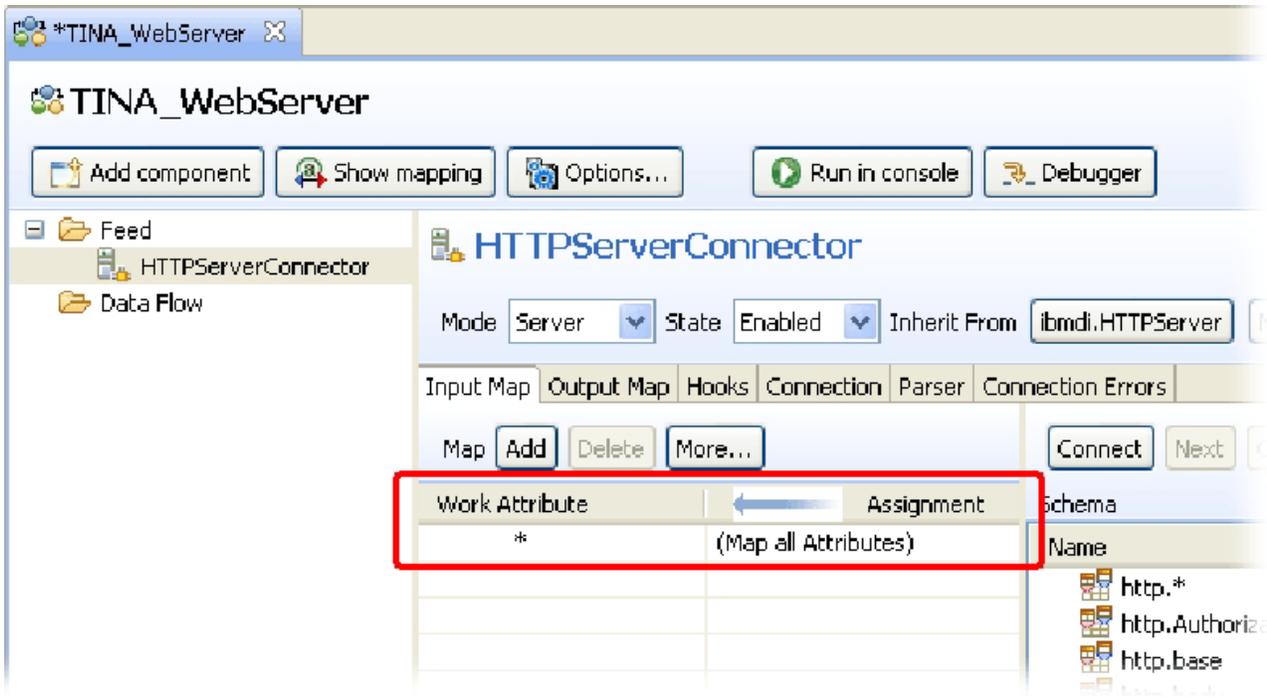


Figura 81. Elemento de correlación comodín

Añada un elemento de correlación comodín también en la correlación de salida, para asegurarse de que los atributos que se establezcan en la entrada de trabajo para el mensaje de respuesta se correlacionarán de vuelta al cliente.

Para probar este componente, ponga un componente de script 'Volcar entrada de trabajo' en la sección Flujo de datos y después **Ejecute** la línea de ensamblaje. La salida del registro debe mostrar un mensaje indicando que el conector de servidor HTTP está a la escucha de conexiones HTTP en el puerto 80³⁴. Esto significa que su LE está esperando que se conecte algún cliente, y esto es lo que debe hacer a continuación, abriendo un navegador y yendo a la siguiente dirección URL:
<http://localhost:80>

Ahora mire la salida del registro. Verá una serie de propiedades de cabecera TCP y HTTP que se han devuelto como atributos.

34. Si, por algún motivo, este puerto ya se está utilizando, abra el panel de configuración del conector de servidor HTTP y seleccione otro puerto.

```

12:54:38,165 INFO - [HTTPServerConnector] CTGDIS498I Using provided parameter
12:54:38,181 INFO - CTGDIS003I *** Start dumping Entry
12:54:38,197 INFO - Operation: generic
12:54:38,197 INFO - Entry attributes:
12:54:38,197 INFO - tcp.localPort (replace): '80'
12:54:38,197 INFO - http.Cookie (replace): 'JSESSIONID=0000MzTM5PekFp
12:54:38,197 INFO - tcp.outputstream (replace): 'java.net.SocketOutput
12:54:38,212 INFO - http.Accept-Language (replace): 'en-us,en;q=0.5'
12:54:38,212 INFO - tcp.localHost (replace): 'localhost'
12:54:38,212 INFO - http.Accept-Charset (replace): 'ISO-8859-1,utf-8;
12:54:38,212 INFO - tcp.inputstream (replace): 'java.net.SocketInputS
12:54:38,212 INFO - http.base (replace): '/'
12:54:38,212 INFO - http.Accept (replace): 'text/html,application/xht
12:54:38,212 INFO - tcp.localIP (replace): '127.0.0.1'
12:54:38,212 INFO - http.Accept-Encoding (replace): 'gzip,deflate'
12:54:38,212 INFO - tcp.remotePort (replace): '1993'
12:54:38,212 INFO - http.User-Agent (replace): 'Mozilla/5.0 (Windows;
12:54:38,212 INFO - tcp.remoteHost (replace): 'localhost'
12:54:38,212 INFO - http.method (replace): 'GET'
12:54:38,212 INFO - http.Host (replace): 'localhost:80'
12:54:38,212 INFO - http.Connection (replace): 'keep-alive'
12:54:38,212 INFO - http.url (replace): '/'
12:54:38,228 INFO - http.Keep-Alive (replace): '300'
12:54:38,228 INFO - tcp.remoteIP (replace): '127.0.0.1'
12:54:38,228 INFO - tcp.socket (replace): 'Socket[addr=localhost/127
12:54:38,228 INFO - CTGDIS004I *** Finished dumping Entry
12:54:38,322 INFO - CTGDIS003I *** Start dumping Entry

```

Figura 82. Propiedades de cabecera TCP y HTTP devueltas como atributos

El único atributo que le interesa para este ejercicio es 'http.base', que contiene la parte del URL que aparece después del host y el socket. Concretamente, comprobará que contiene el texto 'RunAL'.

Para buscar este texto, añada una rama IF a la sección Flujo de datos de la LE. Póngale el nombre 'RunAL detected' y defina la condición: *http.base contiene 'RunAL'*.

Si esta condición de rama evalúa a *verdadero*, queremos que se inicie la LE 'CSV2XML_LookupMode'. Para conseguirlo, deberá reutilizar el componente de función de línea de ensamblaje de la LE anterior 'Scheduler', arrastrándolo primero a **Recursos > Funciones** en el panel del navegador y después a esta línea de ensamblaje, soltándolo encima de la nueva rama IF.

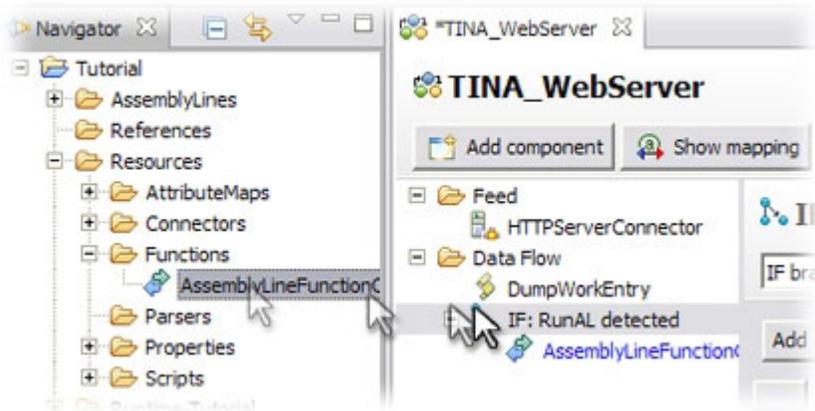


Figura 83. Arrastrar el componente de función de línea de ensamblaje

A continuación, vuelva a ejecutar la línea de ensamblaje y escriba el siguiente texto en el campo de dirección de su navegador:

```
http://localhost/RunAL
```

Verá el volcado de la entrada de trabajo en la salida del registro, seguido de las estadísticas de la línea de ensamblaje a la que se ha llamado.

```

TINA_WebServer x
- http.Base (replace): /RunAL
- http.Accept (replace): 'text/html,application/xhtml+xml,application/xml;q=0.9,
- tcp.localIP (replace): '127.0.0.1'
- http.Accept-Encoding (replace): 'gzip,deflate'
- tcp.remotePort (replace): '2132'
- http.User-Agent (replace): 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv
- tcp.remoteHost (replace): 'localhost'
- http.method (replace): 'GET'
- http.Host (replace): 'localhost'
- http.Connection (replace): 'keep-alive'
- http.url (replace): '/RunAL'
- http.Keep-Alive (replace): '300'
- tcp.remoteIP (replace): '127.0.0.1'
- tcp.socket (replace): 'Socket[addr=localhost/127.0.0.1,port=2132,localport=80
- CTGDIS004I *** Finished dumping Entry
- [AssemblyLineFunctionComponent] CTGDIS255I AssemblyLine AssemblyLines/CSV2XML_LookupMe
- [AssemblyLineFunctionComponent] [Read_CSV_File] CTGDJW003I Parser will use first input
- [AssemblyLineFunctionComponent] CTGDIS087I Iterating.
- [AssemblyLineFunctionComponent] *** Skipping incomplete entry
- [AssemblyLineFunctionComponent] CTGDIS003I *** Start dumping Entry
- [AssemblyLineFunctionComponent] Operation: generic
- [AssemblyLineFunctionComponent] Entry attributes:
- [AssemblyLineFunctionComponent] Last (replace):
- [AssemblyLineFunctionComponent] Title (replace):
- [AssemblyLineFunctionComponent] First (replace): 'Roger'
- [AssemblyLineFunctionComponent] FullName (replace): 'Roger '
- [AssemblyLineFunctionComponent] CTGDIS004I *** Finished dumping Entry
- [AssemblyLineFunctionComponent] CTGDIS088I Finished iterating.
- [AssemblyLineFunctionComponent] CTGDIS100I Printing the Connector statistics.
- [AssemblyLineFunctionComponent] [Read_CSV_File] Get:7
- [AssemblyLineFunctionComponent] [Incomplete data] Branch True:1, Branch False:6
- [AssemblyLineFunctionComponent] [Write to log] (No statistics for script component.)
- [AssemblyLineFunctionComponent] [DumpWorkEntry] (No statistics for script component.)
- [AssemblyLineFunctionComponent] [Exit Flow] (No statistics for script component.)

```

Figura 84. Volcado de la entrada de trabajo seguido de las estadísticas de la LE

El servicio ya funciona, pero el ejercicio no termina aquí. Primero lo haremos más bonito y mucho más fácil de utilizar haciendo que la línea de ensamblaje de servidor web devuelva algunas páginas HTML. Esto, en condiciones normales, requeriría escribir código script. Afortunadamente, dispone de varios archivos de guía de aprendizaje que reducen la tarea a un simple caso de arrastrar y soltar.

Antes de empezar, inhabilite el script 'Volcar entrada de trabajo' para minimizar la salida del registro. Después añada una rama ELSE inmediatamente después de la rama 'IF RunAL detected'. Póngale el nombre 'Return web page'. A continuación, utilice un explorador de archivos para localizar el archivo de script llamado Return web page.script en el directorio Tutorial y arrástrelo a **Recursos > Scripts**. Desde allí, puede arrastrarlo hasta su LE y soltarlo sobre la rama ELSE. Su LE debería ahora tener este aspecto:

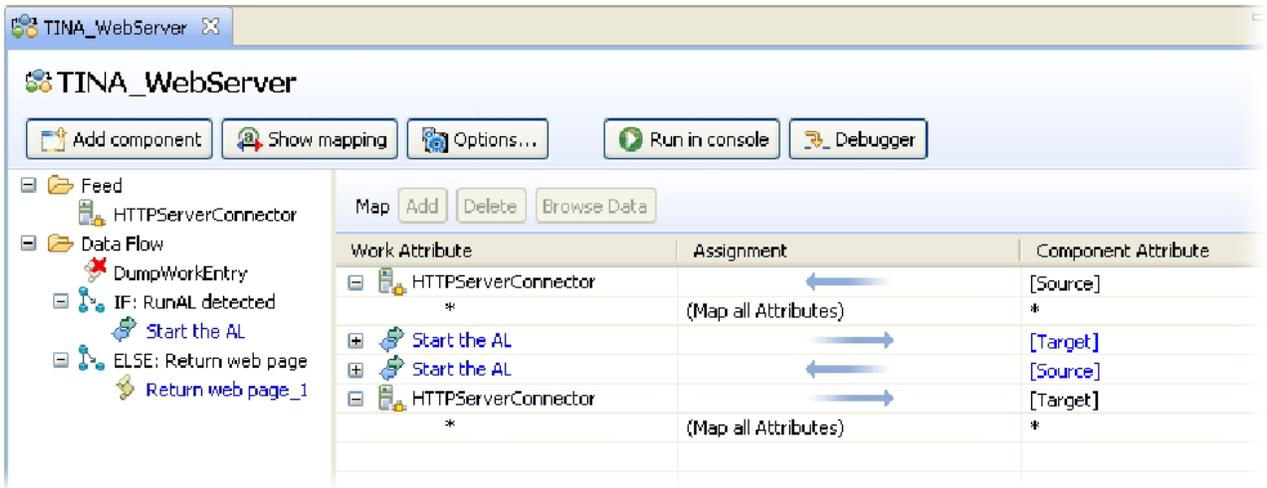


Figura 85. Línea de ensamblaje TINA_WebServer completada

Vuelva a ejecutar la línea de ensamblaje y, cuando vaya a <http://localhost>, debería ver esta página web:

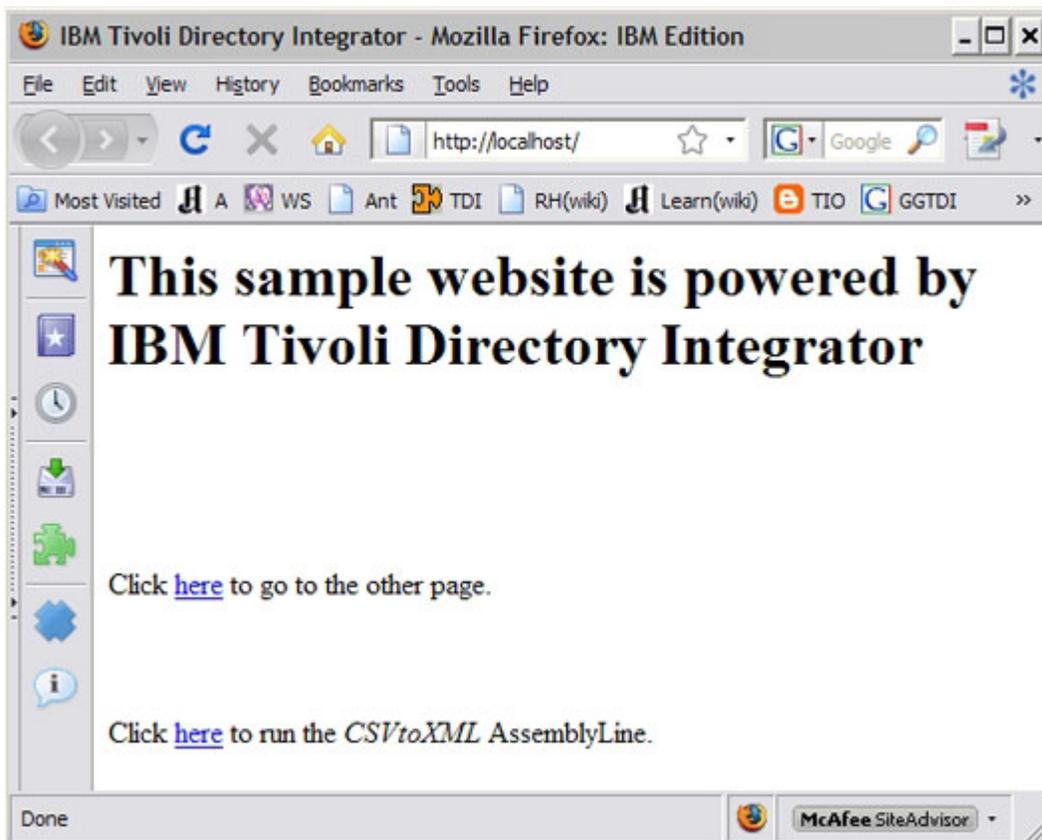


Figura 86. Interfaz web simple a la solución

El primer enlace irá a `OtherPage.html`, mientras que el segundo debería enviar el texto 'RunAL' de `http.base` para lanzar su línea de ensamblaje.

Con esto concluye la parte práctica de esta guía.

Capítulo 4. Fortalecer las soluciones de integración

Como ha visto, crear líneas de ensamblaje puede resultar una empresa relativamente rápida. Sin embargo, una LE finalizada no significa que su solución esté a punto para ser utilizada intensivamente. Incluso las tareas de integración más sencillas requieren un mínimo de reflexión y análisis de resultados.

Algunas preguntas pertinentes sobre las que reflexionar son:

- ¿Se procesan todos los datos de origen de la forma esperada? ¿Cómo puede confirmarlo?
- ¿Se detectan las anomalías en el contenido y/o en la calidad de los datos? ¿Se gestionan?
- ¿El proceso afecta a otros conjuntos de datos, sistemas o líneas de ensamblaje? ¿Cómo?
- ¿La integración supone una carga de auditoría?
- ¿Quién desplegará la solución? ¿Quién la utilizará, y quién la administrará? ¿Cómo?

Para líneas de ensamblaje de larga duración, como las que utilizadas para sincronizaciones y para alimentar servicios, puede considerar otros aspectos, como disponibilidad y tolerancia a fallos, rendimiento, escalabilidad y seguridad.

El objetivo de esta última sección es que tome conciencia de estos problemas y que conozca una serie de características y técnicas de IBM Security Directory Integrator que puede utilizar para abordarlos.

Nota: Si es la primera vez que utiliza IBM Security Directory Integrator, no se preocupe si esta sección le parece compleja o difícil de seguir. Vuelva a leer estas secciones más tarde, cuando tenga más experiencia y se sienta más cómodo utilizando el sistema.

Legibilidad, reutilización y configurabilidad

Todo trabajo de desarrollo requiere resolución de problemas, mantenimiento y ampliación. Las soluciones de IBM Security Directory Integrator no son una excepción.

Puede facilitar estos aspectos siguiendo algunas directrices básicas.

1. Escriba sus líneas de ensamblaje teniendo siempre presente que otras personas tendrán que entenderlas, utilizarlas y mantenerlas. Esto significa intentar que las LE sean lo más cortas posible y que sus componentes tengan nombres claros y descriptivos. Si implementa la lógica del flujo de la LE mediante ramas y bucles, será más fácil de leer y depurar para no programadores que si utiliza scripts 'ocultos' en enganches o empaquetados en componentes de script.
2. A partir de la regla de que la *LE sea lo más corta posible* se deriva que se debe intentar que los fragmentos de código script sean también lo más cortos posible. En vez de escribir bloques monolíticos de código, divídalos en unidades más pequeñas, incluso poniéndolos en componentes de script distintos para mejorar la legibilidad y facilitar la depuración. De esta forma, se puede inhabilitar un componente de script para saltarse parte del código.

Otra forma de mejorar la legibilidad y evitar la duplicación es utilizar la herencia en los componentes de script (desde la carpeta 'Scripts' en la vista de árbol del navegador) y definiendo funciones para las tareas frecuentes. Una línea de ensamblaje se ejecuta en el contexto de su propio motor de scripts, de forma que todas las variables y funciones declaradas en un sitio están disponibles a lo largo de toda la LE. Un lugar donde se suelen declarar estas variables y funciones es en los enganches prólogo de la LE, o en scripts que se hayan seleccionado como "Prólogos adicionales"³⁵.

3. Prefiera la legibilidad a la elegancia en lo que concierne a sus algoritmos, y tenga presente que cuando retome su propio trabajo dentro de seis meses, probablemente será como si lo hubiera hecho otra persona. El hecho de ser considerado con los colegas revertirá en un provecho para usted mismo.
4. Recuerde que es posible que personas que no conozcan el Editor de configuración tengan que modificar los valores y ejecutar sus líneas de ensamblaje. Las LE se pueden iniciar fácilmente desde la línea de mandatos:

```
ibmdisrv -c myConfig.xml -r myAssemblyLine
```

Esto significa que puede preparar scripts o archivos de proceso por lotes para facilitar esta tarea.

5. Será más fácil reconfigurar sus LE si externaliza los valores de los parámetros utilizando propiedades. Las propiedades son pares de clave-valor que se pueden almacenar en archivos o bases de datos y que permitirán reconfigurar su solución desde fuera del Editor de configuración. Para ligar las propiedades a los parámetros de los componentes, pulse en la etiqueta del parámetro y seleccione **Añadir propiedad**.

También se pueden consultar y modificar las propiedades desde los scripts con las llamadas `system.getTDIProperty()` y `system.setTDIProperty()`, lo que le permite hacer que se pueda conmutar fácilmente la lógica personalizada también mediante valores de propiedad externos.

Además, las propiedades se pueden cambiar en un servidor en ejecución utilizando el programa de utilidad de línea de mandatos, `bin/tdisrvctl`, que le permite también iniciar y detener líneas de ensamblaje, consultar el estado y cargar o recargar configuraciones - todo ello sin detener el servidor.

6. Como se ha mencionado antes, si utiliza vías de acceso relativas para los archivos, será más fácil mover la solución a una nueva instalación. Se recomienda que haga sus vías de acceso relativas al directorio desde donde se carga el archivo Config XML. Este directorio es accesible mediante la propiedad `{config.$directory}`, que se puede utilizar para parámetros de una vía de acceso específica utilizando la opción de **Texto con sustitución**; por ejemplo:

```
{config.$directory}/html
```

7. Como se ha dicho anteriormente, pero especialmente cuando cree soluciones que serán desplegadas y utilizadas por otras personas, no espere que estos usuarios tengan conocimientos de IBM Security Directory Integrator. Proporcione archivos de proceso por lotes o scripts para iniciar las líneas de ensamblaje, incluso para las LE de prueba y de validación. Estos archivos pueden simplemente conectarse a los orígenes de datos e informar de si el proceso ha sido satisfactorio o ha habido errores. Recuerde que, para imprimir mensajes en la línea de mandatos de la consola, de forma que los archivos de proceso por lotes o scripts devuelvan información de estado, es mejor utilizar el

35. Los Prólogos adicionales se ejecutan antes de que se invoquen los enganches de prólogo de la propia línea de ensamblaje. Se seleccionan en el panel de valores de la línea de ensamblaje, al cual se puede acceder pulsando el botón derecho del ratón sobre una LE y seleccionando **Valores de línea de ensamblaje...**

método de servidor, `main.logmsg()`, en vez de la versión de LE que ha utilizado en los ejercicios de la guía de aprendizaje: `task.logmsg()`. Este última llamada sólo envía el mensaje al registro.

Éstas han sido algunas indicaciones. Encontrará más en otras publicaciones de IBM Security Directory Integrator y en los grupos de noticias.

Registro cronológico y auditoría

IBM Security Directory Integrator utiliza *log4j* para ofrecer una gestión de registro flexible. Puede elegir entre una serie de *Appenders* estándar, incluyendo *appenders* para el *syslog* de Unix, el registro cronológicos de sucesos de Windows, los archivos diarios y los registros secuenciales. También se pueden crear o descargar nuevos *appenders*.

De forma predeterminada, sólo está habilitado el registro de servidor mínimo. Como mínimo, debería definir el registro para sus líneas de ensamblaje utilizando el *appender* *FileRoller*, escribiendo en la subcarpeta 'logs' de su directorio de soluciones y poniendo al archivo de registro el mismo nombre que el de la línea de ensamblaje. Así pues, para la LE 'CSV2XML' definiría un conjunto de archivos de registro secuenciales basados en esta vía de acceso de archivo: `logs/CSV2XML.log`.

Observe que el método `logmsg()` le permite definir opcionalmente el nivel de registro de su mensaje pasando una de las siguientes palabras clave como primer argumento justo antes de su mensaje de registro: `DEBUG`, `INFO`, `WARN`, `ERROR`, `FATAL`. Los niveles de registro son inclusivos, de modo que `WARN` incluye `ERROR` y `FATAL`, y `DEBUG` significa que se registrarán los mensajes de todos los niveles. Por ejemplo, un mensaje como:

```
task.logmsg("DEBUG", "Updated: " + conn);
```

sólo lo emitirían *Appenders* definidos para el nivel de registro `DEBUG`.

Puede añadir mensajes de auditoría a su solución que se puedan activar y desactivar desde fuera de un servidor en ejecución prefijando llamadas a las llamadas `task.logmsg()` con una sentencia `IF` que compruebe el valor de una propiedad. Por ejemplo, el siguiente fragmento de código script podría aparecer en un ganche 'Flujo de datos - Update Satisfactorio':

```
if (system.getTDIProperty("MyProps","audit").equalsIgnoreCase("true"))
    task.logmsg("DEBUG", "Updated the following data: " + conn);
```

Mediante el programa de utilidad de línea de mandatos `tdisrvctl` para cambiar el valor de la propiedad "audit" en el almacén de propiedades "MyProps", puede activar o desactivar dinámicamente este tipo de mensaje de auditoría para un servidor en ejecución.

En general, es mejor registrar información en exceso que demasiado poca. Aunque tampoco debe inundar la salida del registro. Ya que puede ser difícil encontrar los mensajes de interés en una salida de registro demasiado abarrotada.

Problemas de conectividad

Los problemas de conectividad se pueden dividir en dos categorías generales: errores de inicialización y conexiones perdidas.

De forma predeterminada, todos los componentes lanzan conexiones cuando empieza a funcionar la LE, durante su fase de inicialización. Si, por algún motivo,

alguna conexión falla en este momento, se invoca al enganche 'Prólogo – Después de un error de conexión' que le proporciona un contenedor de scripts desde donde puede enviar alertas o incluso cambiar los valores de los parámetros y volver a intentar la conexión. De forma similar, si se producen errores de conexión durante el ciclo de la LE, se utiliza el enganche 'Flujo de datos – Al perder la conexión' para manejar esta situación.

Además de esta gestión personalizada, los conectores y los componentes de función también ofrecen funcionalidad incorporada de *reconexión* mediante una pestaña **Errores de conexión**. En ella puede indicar al componente que intente restablecer una conexión perdida o, en caso de un problema de inicialización, que continúe intentando configurar la conexión.

En caso de errores de inicialización, normalmente no se habilita la reconexión a menos que experimente problemas recurrentes como tiempos de espera excedidos durante la negociación SSL o problemas de conexión esporádicos similares.

Sin embargo, se recomienda habilitar **Reconexión automática si se pierde la conexión**, lo que permite a su componente restablecer la conexión y continuar como si no hubiera pasado nada. Tenga en cuenta que en caso de los conectores de modalidad *Iterador*, reconectar significa también que el 'cursor' de iteración se podrá restablecer, de forma que el ciclo empiece otra vez en la primera entrada del conjunto de resultados. Por supuesto, para los iteradores que detectan el estado, como los conectores de detección de cambios, esto no supone ningún problema porque estos componentes utilizan automáticamente información de estado para reanudar allí donde lo han dejado.

Disponibilidad de las líneas de ensamblaje

Mejorar la disponibilidad de las LE significa dos cosas: 1) hacer lo que pueda para asegurarse de que sus líneas de ensamblaje no se detengan y 2) reiniciar las LE que se hayan detenido lo antes posible. En casos como migraciones y sincronizaciones de larga duración, es posible que se necesite también que las líneas de ensamblaje reiniciadas continúen en el punto donde estaban en el momento del error.

Una línea de ensamblaje se detendrá si se produce una excepción no gestionada. Puede evitar esto tratando de gestionar todos los posibles errores, lo cual, en términos de IBM Security Directory Integrator, significa habilitar al menos el enganche 'Valor predeterminado en caso de error' de todos los conectores y componentes de función. Al habilitar los enganches de error, indica al servidor que continúe a pesar de haber encontrado una excepción.

Como se ha visto durante los ejercicios de aprendizaje, si una línea de ensamblaje se detiene a causa de un error, obtiene un rastreo de pila precedido de información sobre dónde ha ocurrido el error y porqué. Si evita que la LE se detenga habilitando enganches de error, es responsabilidad suya informar del estado de error utilizando objetos especiales que tiene disponibles para sus scripts.

```
task.logmsg("ERROR", "[" + thisConnector.getName() + "] - " +  
error);
```

Este script de ejemplo utiliza dos de estos objetos: `thisConnector` que siempre hace referencia al componente al que el script en ejecución está ligado, y la variable predefinida denominada `error`. El objeto `error` es una Entrada, como `work` y `conn`,

y contiene atributos como 'status', 'connectorname'³⁶ y 'message', más otros detalles relevantes sobre cualquier situación reciente de error. Dado que es un objeto Entrada, puede utilizar `task.dumpEntry(error)` para mostrar sus contenidos, así como referencias directas a nombres de atributo - por ejemplo: `work.message` - o puede simplemente añadir `error` a un mensaje como en el ejemplo anterior, ya que todos los objetos Entrada se pueden convertir a representaciones en string según sea necesario.

Para manejar excepciones que tengan lugar en scripts, como en asignaciones de correlaciones de atributos, enganches y componentes de script, escriba su código en bloques try-catch. Esto le permite capturar las excepciones y tratarlas usted mismo:

```
try {
    res = myLib.callToSomeFunctionThatMightFail();
} catch (excptn) {
    task.logmsg("Call failed with error: " + excptn);
}
```

Además de manejar errores quizá desee utilizar la característica de Reconexión automática descrita en la sección anterior. Esto evitaría que la línea de ensamblaje falle debido a problemas de conectividad transitorios, como exceder el tiempo de espera para un origen de datos o cortafuegos.

Si por algún motivo su línea de ensamblaje aún se detiene prematuramente, el próximo paso es reiniciarla. Una forma de hacerlo es utilizar la herramienta de administración web para definir comportamientos de error/respuesta.

Otra forma de hacerlo, que incluso es complementaria, es crear una LE 'lanzadora' y aprovechar el componente de función de línea de ensamblaje que ha utilizado en los ejercicios de aprendizaje de la sección anterior. Poniendo este componente de función de línea de ensamblaje en un bucle condicional que no se detenga nunca - en otras palabras, con una condición que siempre dé *verdadero* - y configurando después el componente de función de línea de ensamblaje para que llame a la LE deseada y espere a que se complete, se asegura de que cuando el control vuelve a la 'LE lanzadora', el bucle interminable simplemente reiniciará otra vez el flujo de datos.

Si aplica la técnica de bucle de reinicio indicada arriba para una LE de sincronización basada en uno de los conectores de detección de cambios (o el motor delta, ambos descritos en la bibliografía) la línea de ensamblaje continuará automáticamente en el punto donde haya fallado. Si no, el trabajo de manejar el estado queda a su cargo. Una técnica frecuente es utilizar el almacén del sistema para conservar la información de estado, como indicaciones de fecha y hora y otros valores clave para conjuntos de resultados ordenados. Así, cuando la línea de ensamblaje se inicialice, esta información de estado se aplica al conector iterador para reanudar el proceso inmediatamente después de la entrada anterior gestionada.

Si el origen de datos de iteración de la LE no admite devoluciones ordenadas, quizás sea necesario iniciar la iteración de nuevo desde el principio. En tal caso, observe que la modalidad de Actualización del conector ofrece una característica

36. Habrá observado que la palabra 'connector' a veces es sinónimo de 'component', de forma que variables como `thisConnector` también pueden hacer referencia a componentes de función, componentes de script y componentes de correlación de atributos. Lo mismo se aplica al atributo 'connectorname' del objeto error, que también puede contener el nombre de cualquier tipo de componente.

para **Calcular cambios** que compara los atributos correlacionados de salida con los que encuentra actualmente en el sistema de destino, saltándose la operación de modificar si no detecta diferencias.

Puede evitar un único punto de anomalía introduciendo un transporte seguro entre diversos servidores de IBM Security Directory Integrator, como IBM MQ. De esta forma, se puede utilizar cualquier número de servidores /y líneas de ensamblaje) para iniciar el proceso poniendo datos e incluso instrucciones de proceso en la cola. En el extremo de recepción, diversos servidores/líneas de ensamblaje los cogen en orden según el principio 'primero en llegar, primero en ser atendido' y llevan a cabo el trabajo solicitado. Esto no sólo da como resultado una solución más robusta, sino que además permite ampliarla añadiendo líneas de ensamblaje remitentes y destinatarias.

En esta sección hemos tratado brevemente un tema que es mucho más amplio. Sin embargo, el objetivo es más proponer ideas que indicar una forma de trabajo determinada. Se recomienda que eche un vistazo a los sitios web de comunidad y en los grupos de discusión para encontrar ejemplos y recomendaciones más concretas.

Ampliación y rendimiento

Este tema también merece una discusión más larga de la que encontrará aquí. Sin embargo, vale la pena mencionar algunos puntos, aunque sea de forma general.

El factor principal para controlar la velocidad de sus flujos de datos es la E/S; el tiempo que se necesite para recuperar datos de los orígenes de datos o para ponerlos en los destinos finales eclipsa en gran medida el tiempo de proceso dentro de la propia LE. Además, negociar las conexiones puede ser también muy costoso, especialmente cuando también implica reconocimiento de seguridad. Si tiene esto presente cuando diseñe y cree sus líneas de ensamblaje, tendrá un impacto directo en el rendimiento de las mismas.

Por ejemplo, imaginemos una LE basada en modalidad servidor que reciba solicitudes de protocolo entrantes de clientes, como la solución de servidor HTTP que ha creado en el último ejercicio. Cada solicitud recibida hace que se lance una línea de ensamblaje de servicio para llevar a cabo el trabajo real. Si esta línea de ensamblaje invocada tiene que inicializar componentes, el tiempo de respuesta para cada solicitud será por lo menos tan largo como la suma de todos los tiempos de conexión.

Tres formas de paliar esta situación serían:

- Hacer que la LE principal en modalidad servidor realice el proceso real en vez de enviarlo a otra LE, con lo cual no sería necesario establecer e interrumpir las conexiones para cada solicitud;
- Diseñar la(s) línea(s) de ensamblaje de servicio de forma que se puedan invocar en *Modalidad manual/ciclo*. La modalidad manual/ciclo hace que la LE de servicio se inicialice cuando se inicializa el Componente de función de línea de ensamblaje. Además, el componente de función de línea de ensamblaje, dirige a la LE de servicio sólo un único ciclo para cada llamada. De esta forma, la LE de servicio actúa sólo como un componente de la línea de ensamblaje que la llama y se debe construir para que se adapte a este comportamiento;

- Utilizar la agrupación global de conectores. Esta característica se describe en *Referencia* y le permite definir una agrupación de conectores que se inicialicen cuando se inicie el servidor y se compartan entre diferentes líneas de ensamblaje según los necesiten.

Otra forma de mejorar el rendimiento es dividir las tareas de proceso entre diversas líneas de ensamblaje simultáneas. Por ejemplo, imagine una tarea de migración donde en vez de tener una línea de ensamblaje que trabaje con todo el conjunto de datos, lance diversas líneas de ensamblaje cada una de las cuales trate un subconjunto de los datos. Puesto que puede pasar parámetros de inicialización a una línea de ensamblaje al llamarla, puede desarrollar una LE que se inicie diversas veces y que contenga un parámetro de filtro para controlar el rango de datos que esa instancia debe procesar.

Otra técnica sería incorporar un bus de mensajes en la solución, como se ha descrito en la sección anterior. Esta técnica se ha utilizado de forma muy satisfactoria para algunos de los clientes más importantes de IBM.

En casos en que la velocidad de proceso se vea mermada por enlaces de red inestables o sistemas con una baja disponibilidad, puede desplegar líneas de ensamblaje adicionales como tareas en segundo plano para sincronizar datos que sean difíciles de alcanzar con almacenes locales de alta velocidad. Concretamente, para implementar servicios en tiempo real, esta técnica puede garantizar un tiempo de respuesta satisfactorio para las solicitudes de clientes.

Supervisión

Las soluciones de IBM Security Directory Integrator se pueden administrar de forma rápida y sencilla con la herramienta de administración web. Este conector de la Integrated Service Console (ISC) es una aplicación de AppServer que puede supervisar cualquier número de soluciones que se ejecuten en cualquier número de servidores dentro de su infraestructura. Además de las características para visualizar las estadísticas de la LE, los registros y las horas de inicio/detención, la herramienta de administración web le permite personalizar la consola de estado e incidencias, así como definir planificaciones y el comportamiento de error/respuesta para mantener sus líneas de ensamblaje en funcionamiento.

También puede configurar sus soluciones de IBM Security Directory Integrator para enviar sucesos de estado, por ejemplo en forma de interrupciones SNMP o utilizando el formato de sucesos personalizado del sistema. El sistema es fácilmente configurable para la administración y supervisión JMX, utilizando, por ejemplo ITM.

El depurador de líneas de ensamblaje

Aunque se ha mencionado anteriormente, vale la pena repetirlo: el tiempo que invierta en aprender a utilizar el depurador de líneas de ensamblaje lo compensará multiplicado por diez ya que podrá desarrollar soluciones, desplegarlas y solucionar problemas mucho más rápido.

Apéndice. Guía de EasyETL

'ETL' significa Extraer, Transformar y Cargar, y se reduce a obtener datos de un lugar, modificándolos si es necesario, y a ponerlos en otro lugar. EasyETL es una característica de IBM Security Directory Integrator que le permite hacer esto de forma rápida e interactiva con sólo pulsar unas teclas.

Estos son algunos ejemplos de ETL comunes:

- Exportar registros de base de datos, documentos de notas, entradas de directorio o incluso correo entrante o mensajes MQ a un archivo;
- Cargar datos de un archivo a un sistema o almacén de datos
- Migrar datos directamente de un sistema a otro, o entre versiones del mismo sistema en el caso de actualizaciones de software/esquema

IBM Security Directory Integrator EasyETL le permite abordar estas y otras situaciones siguiendo unos pocos pasos intuitivos que producen soluciones adecuadas tanto para un movimiento de datos ocasional como para flujos de datos esenciales en la infraestructura.

El primer paso para crear un nueva tarea EasyETL es elegir el origen de entrada y seleccionar los atributos que desea transferir. En este momento, IBM Security Directory Integrator ya le permite ejecutar el trabajo de EasyETL y colocar los datos leídos en el almacenamiento intermedio de copia para pegarlos. Si los datos se deben transformar o calcular, EasyETL le permite añadir una transformación, volver a ejecutar el trabajo ETL y copiar/pegar los datos transformados. Y también puede elegir un destino de salida y establecer que el trabajo EasyETL escriba los datos directamente ahí.

Una vez que la solución EasyETL funcione del modo deseado, IBM Security Directory Integrator puede generar los activos de línea de mandatos (archivos de proceso por lotes o scripts) para iniciar y planificar la tarea de integración. Finalmente, EasyETL aprovecha las funciones de detección de cambios de IBM Security Directory Integrator para convertir rápidamente el trabajo ETL en una tarea de sincronización de datos.

Nota: La buena noticia para los usuarios de IBM Security Directory Integrator es que cada solución EasyETL es un proyecto de IBM Security Directory Integrator con una sola línea de ensamblaje, y se puede abrir en el entorno de desarrollo con funciones completas. Sin embargo, una vez que lo cambie, ya no estará disponible como EasyETL.

Utilización de EasyETL

Inicie el editor de configuración de IBM Security Directory Integrator y seleccione el espacio de trabajo que se debe utilizar. Cuando IBM Security Directory Integrator se inicia por primera vez se abre en la página de bienvenida³⁷.

37. Puede volver a la página de bienvenida en cualquier momento seleccionando **Ayuda > Bienvenido** en el menú principal

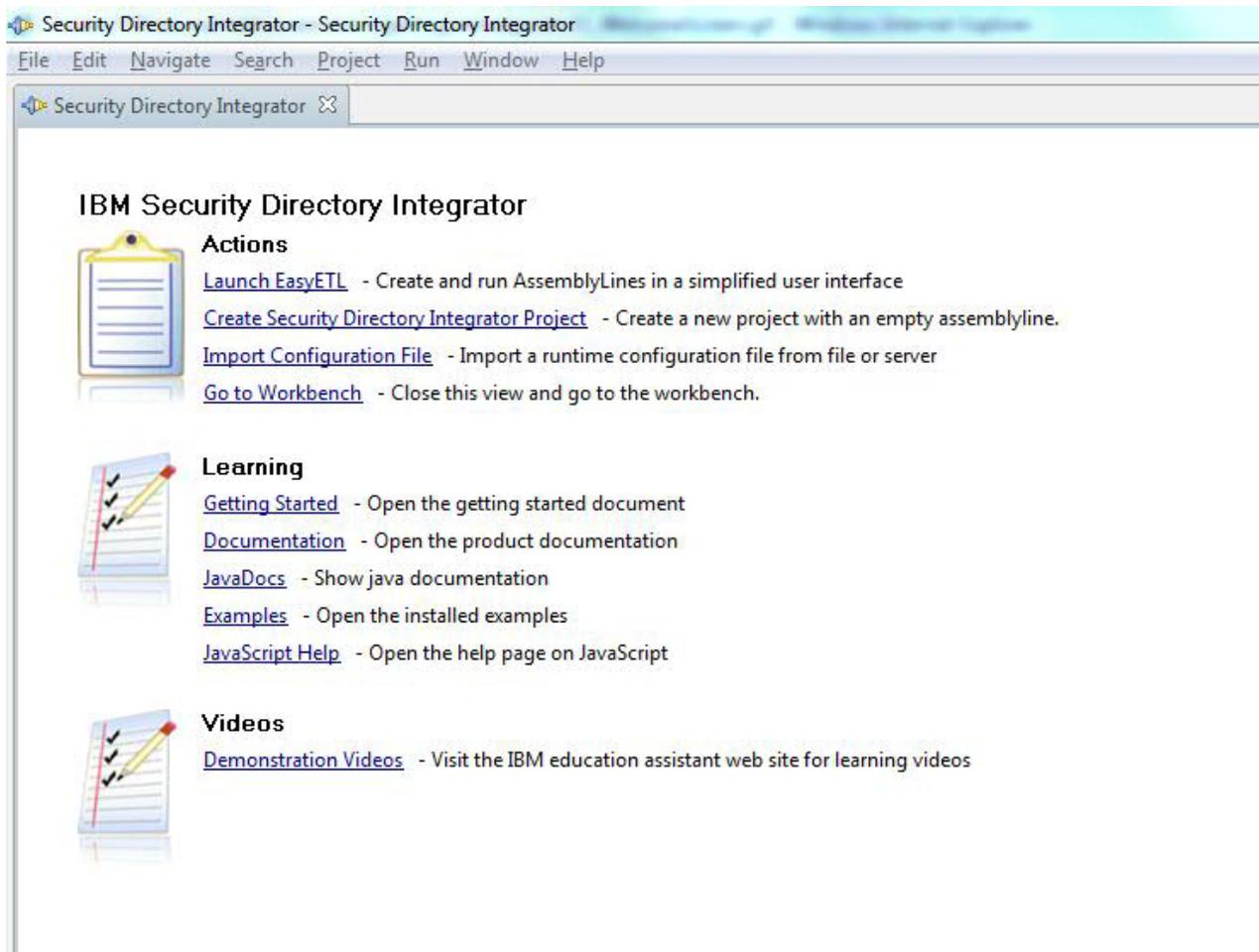


Figura 87. Pantalla de bienvenida

Como se muestra en la imagen anterior, el enlace situado en la parte superior abre el entorno de trabajo EasyETL³⁸. Pulse este enlace para abrir el entorno de trabajo Security Directory Integrator EasyETL.

38. EasyETL es una *perspectiva* de Security Directory Integrator, y se puede alternar entre perspectivas utilizando la selección de menú: **Windows > Abrir perspectiva > Otros...** Si ha realizado cambios en una perspectiva y desea restablecerla en el valor predeterminado, simplemente seleccione **Windows > Restablecer perspectiva**

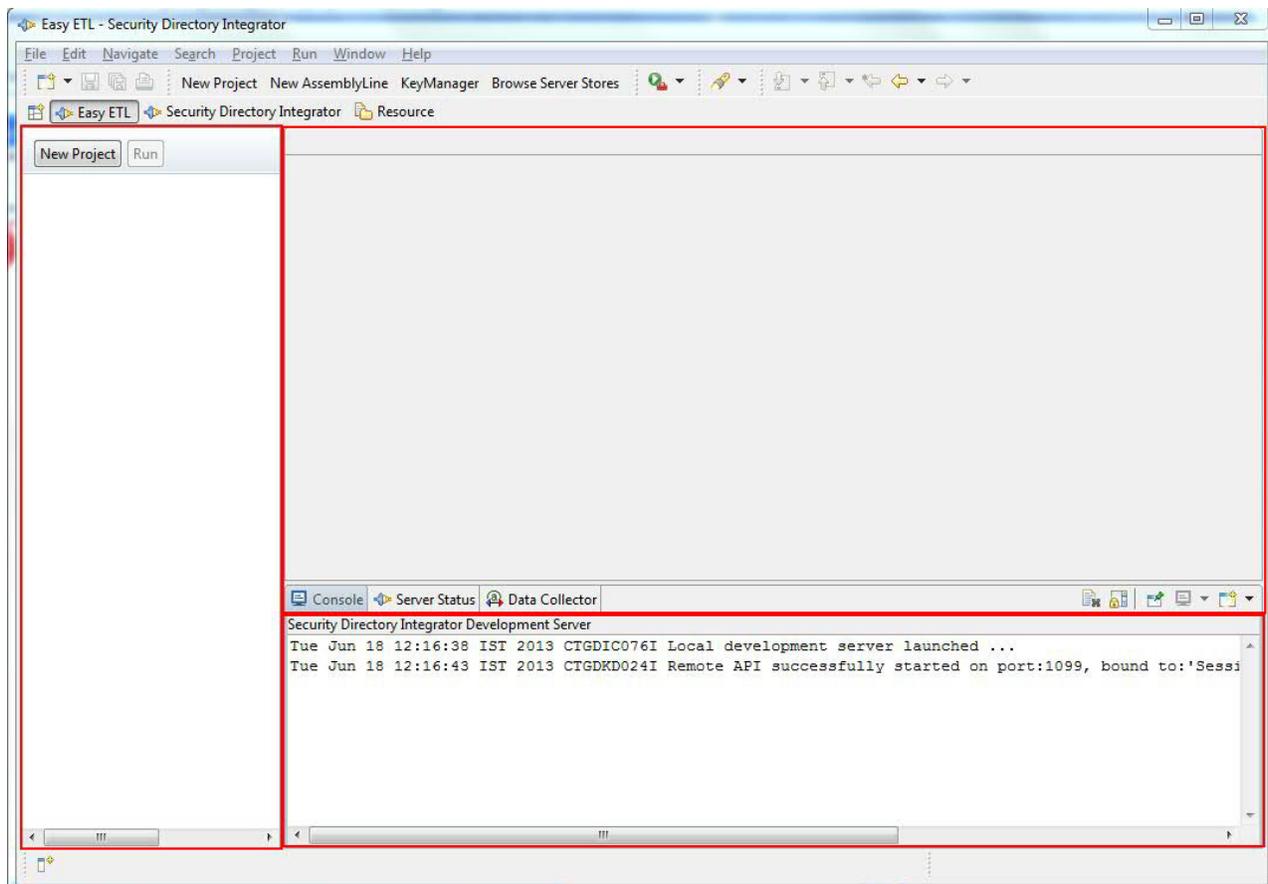


Figura 88. Figura 2. Entorno de trabajo de EasyETL

El entorno de trabajo de EasyETL muestra tres cosas:

- El navegador de proyectos que lista los trabajos ETL. Puede pulsar el botón derecho del ratón en cualquier proyecto para, por ejemplo, ejecutarlo o crear activos de línea de mandatos para iniciarlo
- Un editor de línea de ensamblaje simple³⁹ para cada proyecto abierto
- Varias vistas como pestañas en la parte inferior de la pantalla. De forma predeterminada, se obtienen tres vistas:
 - La salida de consola procedente del servidor IBM Security Directory Integrator de prueba
 - La vista de estado del servidor, donde puede supervisar tanto el servidor como los proyectos EasyETL en ejecución
 - Y el recopilador de datos, donde los datos resultantes para cada ciclo se muestran en una lista tabular

Hallará más información sobre cada sección a medida que avance en este documento.

39. Una línea de ensamblaje es la implementación de un flujo de datos en IBM Security Directory Integrator, por lo tanto, cuando crea o abre un proyecto EasyETL, la línea de ensamblaje subyacente aparece en el editor.

Observe también que el término 'línea de ensamblaje' aparece abreviado como 'LE' en este y otros documentos de IBM Security Directory Integrator.

Creación de un proyecto

Cree un nuevo proyecto EasyETL pulsando el botón **Nuevo proyecto** situado en la parte superior del navegador de proyectos.

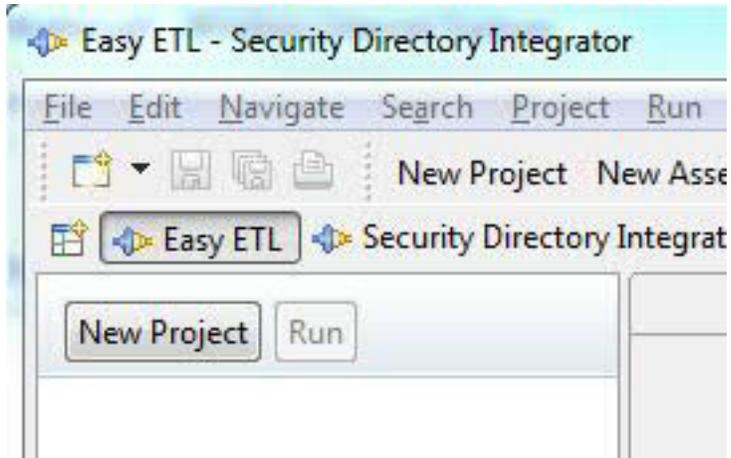


Figura 89. Botón Nuevo proyecto

Asigne a este proyecto el nombre 'CSVtoXML' y pulse **Finalizar**. Se abrirá el nuevo proyecto en el editor de línea de ensamblaje simple.

Figura 90. Editor de línea de ensamblaje simple

El editor proporciona dos desplegados: uno para seleccionar el origen de la entrada y otro para el destino. El área de debajo está vacía (aparte del texto de ayuda) hasta que se elige el origen.

Configuración de la entrada para la LE ETL

Configure la entrada de línea de ensamblaje EasyETL al pulsar el menú desplegable de Información de origen y al seleccionar **Conector de archivos**.

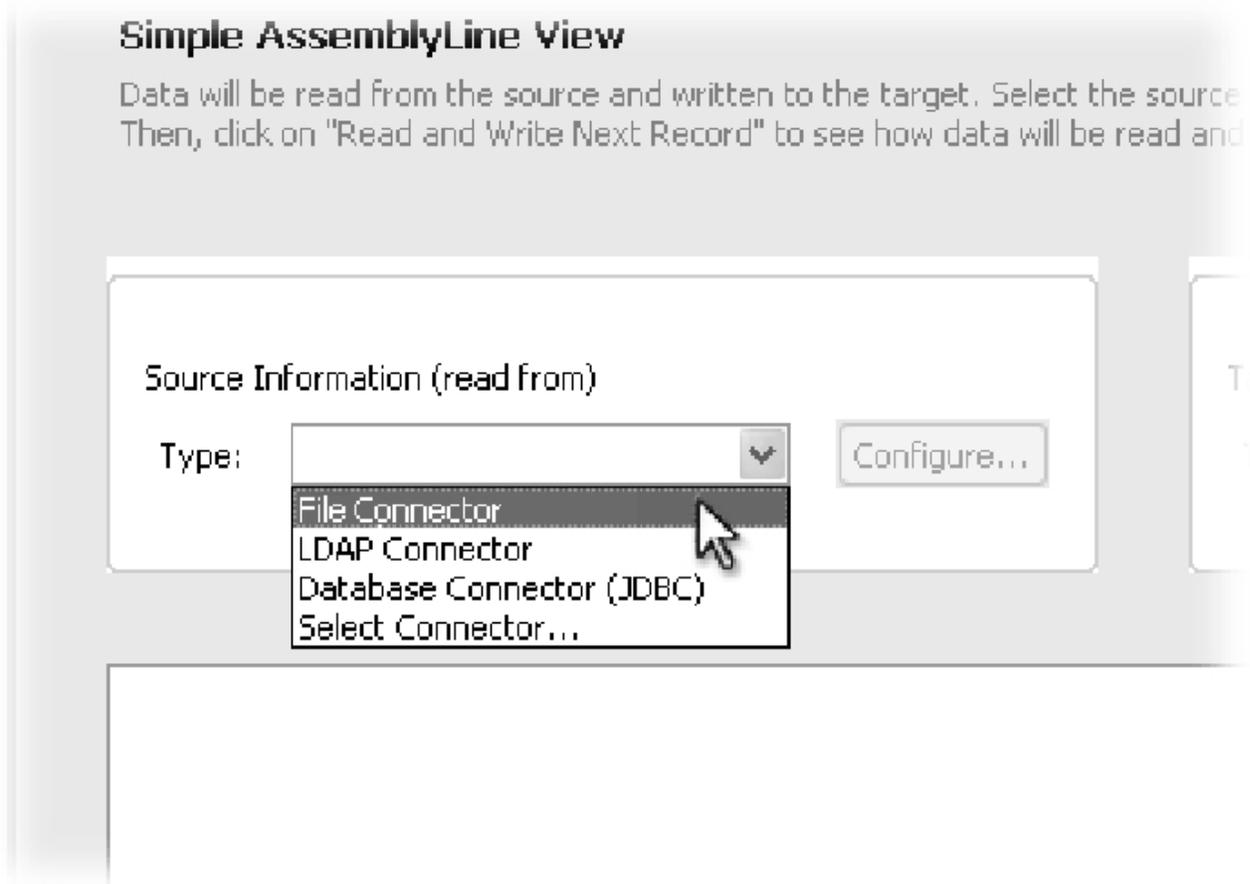


Figura 91. Selección de información de origen

Después aparecerá el diálogo de configuración de este conector.

Apunte el parámetro Vía de acceso del archivo hacia el archivo 'People.csv' que se encuentra aquí:

`TDI_HOME/examples/Tutorial`

donde `TDI_HOME` se sustituye por el directorio de instalación de IBM Security Directory Integrator existente en la máquina⁴⁰.

40. Observe que IBM Security Directory Integrator permite utilizar la barra inclinada y la barra inclinada invertida como separadores en la vía de acceso cuando se ejecuta en Windows. Las soluciones de IBM Security Directory Integrator se podrán migrar más fácilmente entre Windows y el resto de plataformas en las que se ejecuta IBM Security Directory Integrator si utiliza la barra inclinada.

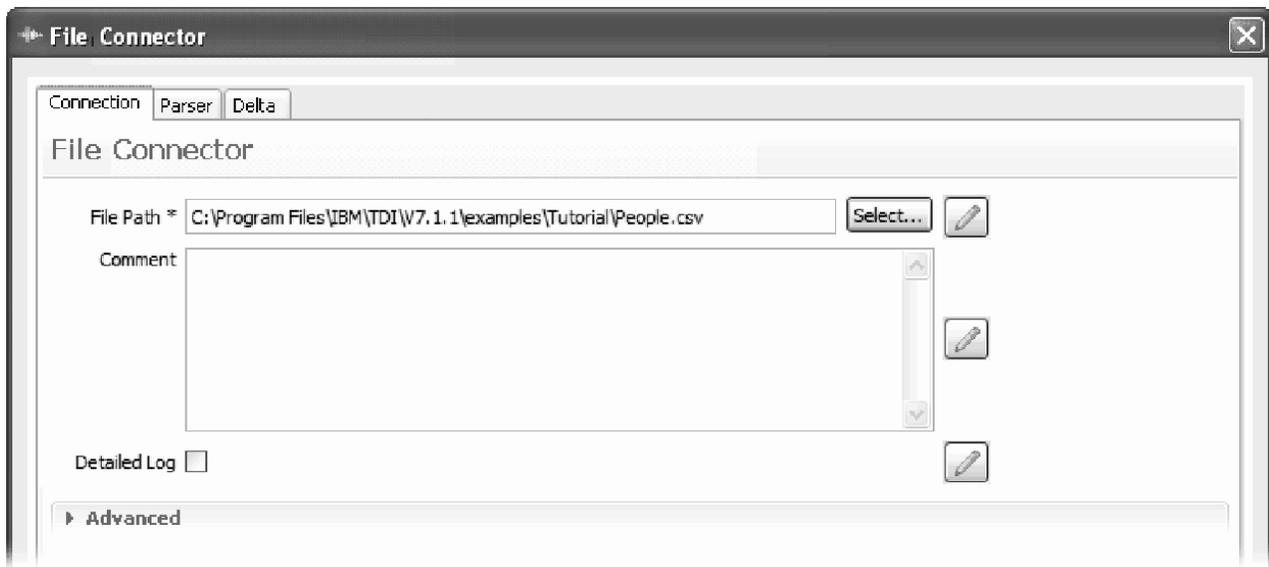


Figura 92. Establecimiento del parámetro Vía de acceso del archivo

Ahora, pulse en la pestaña llamada **Analizador** y seleccione el **Analizador CSV**, dejando tal cual los parámetros de configuración predeterminados. Finalmente, pulse el botón **Conectar** situado en la parte inferior del cuadro de diálogo para probar la conexión y descubrir atributos disponibles.

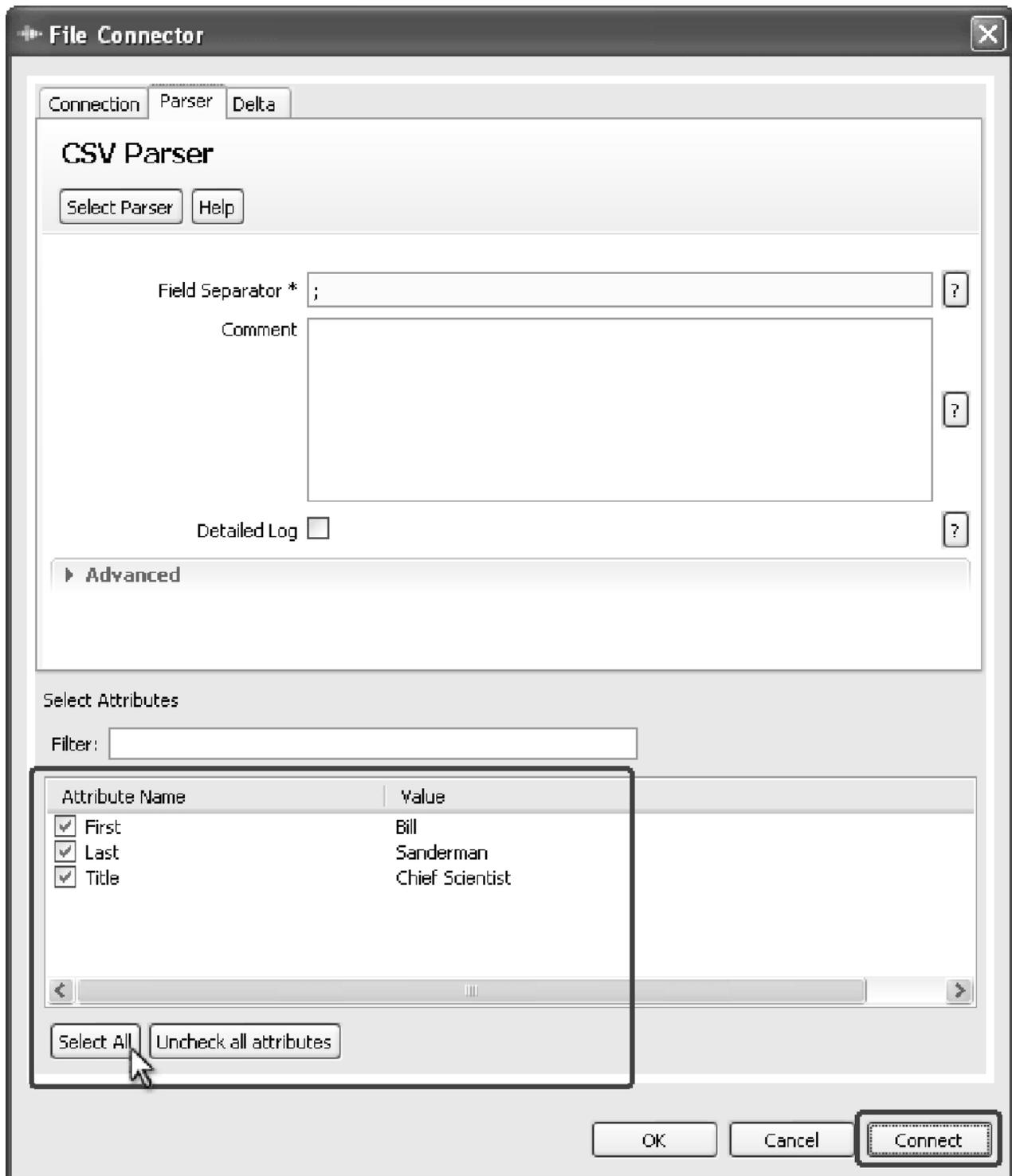


Figura 93. Prueba de la conexión y descubrimiento del esquema

Se muestra el esquema del sistema conectado y a partir de aquí puede elegir cual de estos desea utilizar en el flujo de datos. Para este ejemplo, utilice el botón **Seleccionar todo** y pulse **Aceptar** para cerrar el diálogo de configuración.

De vuelta en el entorno de trabajo EasyETL verá que la mitad inferior del editor de línea de ensamblaje simple ha cambiado para reflejar que ahora tiene un origen de

entrada configurado. IBM Security Directory Integrator ahora le ofrece un botón para recorrer esta información paso a paso, un registro cada vez, así como botones para ejecutar la tarea de ETL hasta su finalización y para detenerla.

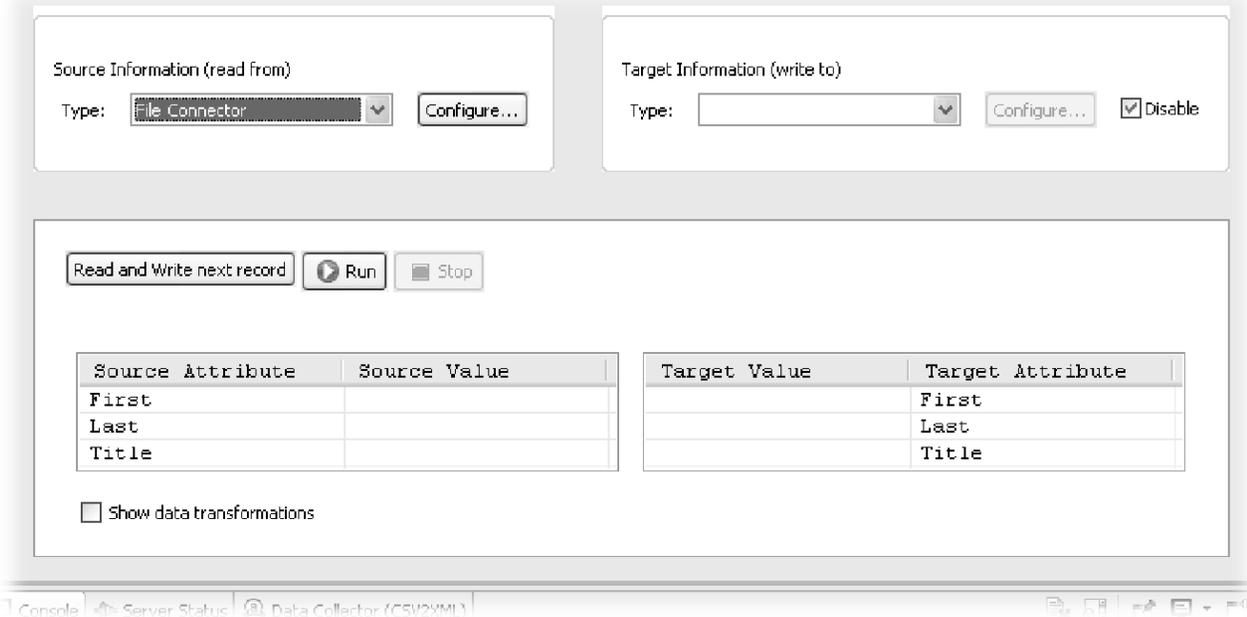


Figura 94. Origen de entrada configurado

Debajo de estos botones hay dos cuadros de cuadrícula llamados *Visores de datos* que listan los atributos gestionados por el flujo de datos. El Visor de datos de la izquierda muestra los atributos de entrada. Los atributos que ha seleccionado para leer en el paso anterior se muestran en negrita en la parte superior y constituyen la *Correlación de entrada*. Debajo se mostrarán los atributos no seleccionados en gris, y puede incluirlos para la correlación de entrada efectuando una doble pulsación sobre ellos. Del mismo modo, los atributos se eliminan de la correlación efectuando una doble pulsación sobre ellos o suprimiéndolos⁴¹.

El recuadro de la derecha es el Visor de salida, y muestra el conjunto de atributos que se van a escribir. En términos de IBM Security Directory Integrator, esto es la *Correlación de salida* y, de forma predeterminada, es idéntica a la lista que ha seleccionado para la Correlación de entrada. Observe que puede cambiar el nombre de cualquier atributo de salida pulsando en la columna de la derecha y editando el valor. Utilice esta técnica para cambiar el nombre del atributo llamado 'Title' por 'JobTitle'.

41. También puede pulsar con el botón derecho del ratón y elegir **Añadir atributo** o **Eliminar atributo**

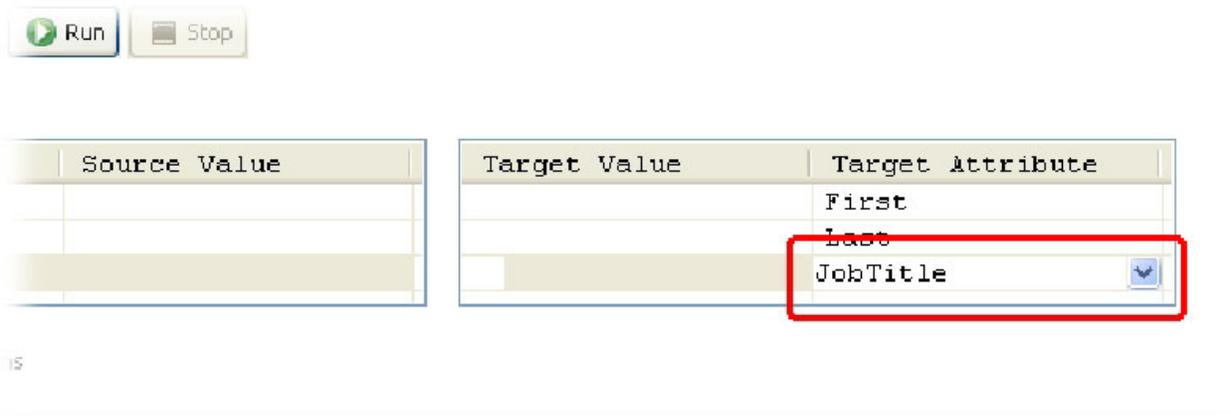


Figura 95. Renombrar un atributo de salida

El proyecto EasyETL está ahora preparado para la primera prueba.

Ejecución de la línea de ensamblaje EasyETL

Seleccione la vista **Recopilador de datos**, situada en la parte inferior de la pantalla, y después pulse el botón **Leer y grabar siguiente registro**. Esto provoca que ocurra lo siguiente:

1. Hay un retraso debido a que la línea de ensamblaje EasyETL se transfiere al servidor en ejecución y se inicia
2. El primer registro se lee y analiza desde el origen de entrada CSV y los datos se muestran en las pantallas de cuadrícula de entrada y de salida
3. Los atributos que ha seleccionado para la salida se escriben y recopilan en la vista Recopilador de datos⁴².

Por lo tanto, aunque no haya seleccionado todavía un destino de salida, todavía puede ejecutar y probar el proyecto ETL, visualizando los datos como fluyen por la línea de ensamblaje.

42. El botón que aparece en el extremo izquierdo de la parte superior de la vista **Recopilador de datos** abre un diálogo de configuración en el que se puede aumentar el tamaño del **almacenamiento intermedio del recopilador de datos**. Tenga en cuenta que si piensa recopilar grandes cantidades de datos, puede ser necesario aumentar la memoria disponible para IBM Security Directory Integrator. Para hacer esto, localice el archivo de proceso por lotes/script 'ibmditk' en la carpeta de instalación de IBM Security Directory Integrator y ábralo en un editor. Cerca de la parte inferior está la línea que inicia 'miadmin', y se puede insertar el siguiente texto después de la opción `-vmargs: -Xmx512M` Este ejemplo permite que la memoria de datos IBM Security Directory Integrator aumente hasta 512 Mbytes.

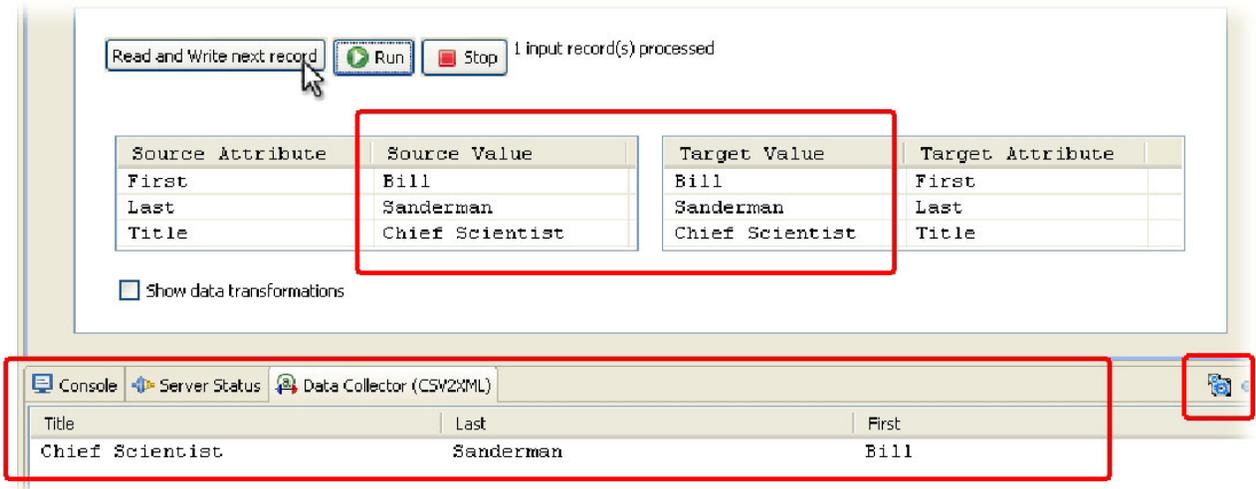


Figura 96. Un registro leído y recopilado

Cada vez que pulsa el botón **Leer y grabar...**, se lee, muestra y recopila otro registro. Si ahora pulsa el botón **Ejecutar**, el trabajo de ETL se ejecuta hasta su finalización y podrá ver este informe LE.

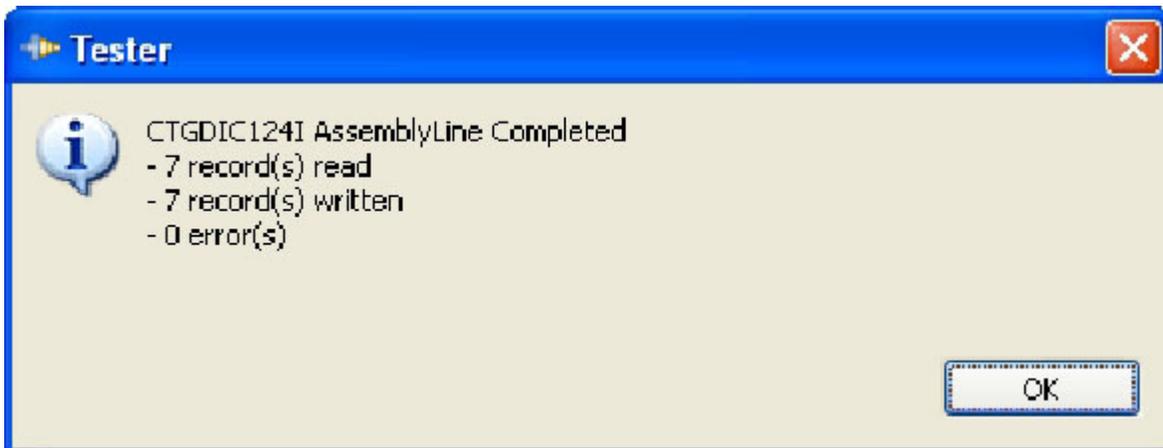


Figura 97. Línea de ensamblaje EasyETL completada

Como se muestra en el diálogo anterior, en realidad no se ha escrito ningún registro en ningún sitio. Sin embargo, el **recopilador de datos** todavía proporciona comentarios visuales útiles sobre la información que se extrae y transfiere.

Además, puede seleccionar filas en los datos recopilados y copiar/pegar esta información en un archivo o en otro destino⁴³.

Transformaciones

Hasta ahora, los valores de salida han sido idénticos a la entrada. No obstante, habrá situaciones en las que desee manipularlos o incluso calcularlos en función de

43. Los datos se copian en formato CSV para simplificar la importación de valores en hojas de cálculo y tablas de informe.'CSV' significa valor separado por caracteres (Character Separated Value). El carácter separador utilizado por IBM Security Directory Integrator es el punto y coma (;).

los datos leídos. Eso se hace en IBM Security Directory Integrator EasyETL escribiendo Transformaciones en JavaScript.

Para poder trabajar con transformaciones, primero debe habilitarlas seleccionando la casilla de verificación **Mostrar transformaciones de datos**.

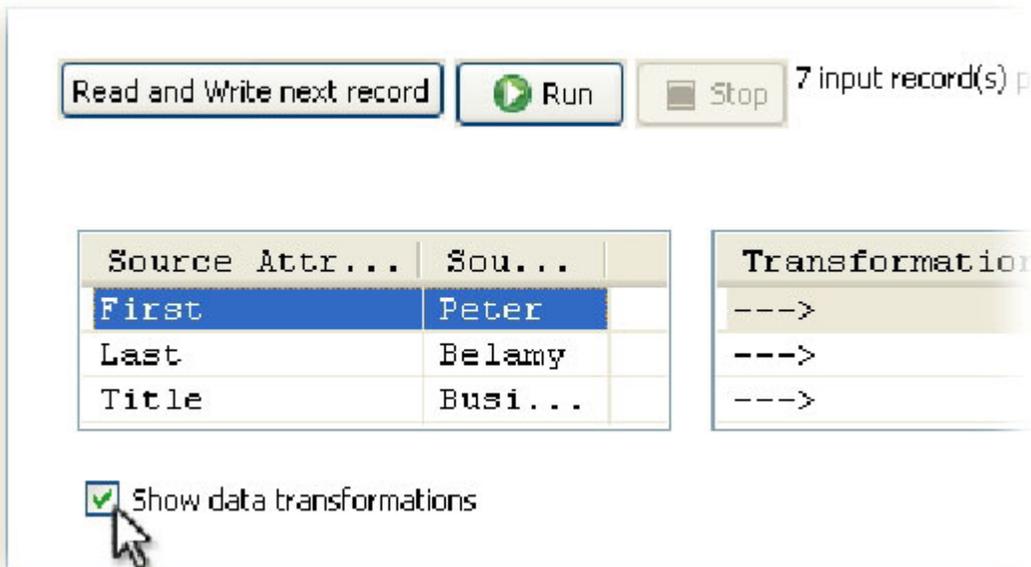


Figura 98. Habilitación de la transformación

Esto provoca la aparición de un nuevo recuadro de cuadrícula entre los visores de datos: el Visor de transformación. Aquí podrá ver flechas que indican que los tres atributos de salida obtienen sus valores directamente de los atributos de entrada, en otras palabras, sin transformaciones. Definirá un nuevo atributo de salida y después añadirá el script de transformación para calcular su valor. Para hacer esto, pulse el botón derecho del ratón en la correlación de salida, seleccione Añadir atributo y asígnele el nombre 'FullName'. Después, efectúe una doble pulsación en la transformación que aparece a la izquierda de este nuevo atributo de salida y después introduzca este fragmento de código script:⁴⁴ `return First + " " + Last`

44. Tenga en cuenta que puede pulsar Ctrl + espacio para obtener una lista de sugerencias de lo que escribe. Esta lista incluye algunos objetos especiales, como 'task' y 'main', así como los atributos que se leen del origen de entrada en la parte inferior.

Tenga también en cuenta que la notación que aquí se muestra para acceder a los atributos de entrada sólo funciona en EasyETL. En el entorno de trabajo completo de IBM Security Directory Integrator, un nombre de atributo debe tener como prefijo el objeto de entrada donde reside el atributo, por ejemplo, la entrada de trabajo (trabajo). Por lo tanto, el ejemplo anterior será similar a esto:

```
return work.First + " " + work.Last
```

La entrada de trabajo y otros conceptos de IBM Security Directory Integrator se describen en la primera sección de este documento.

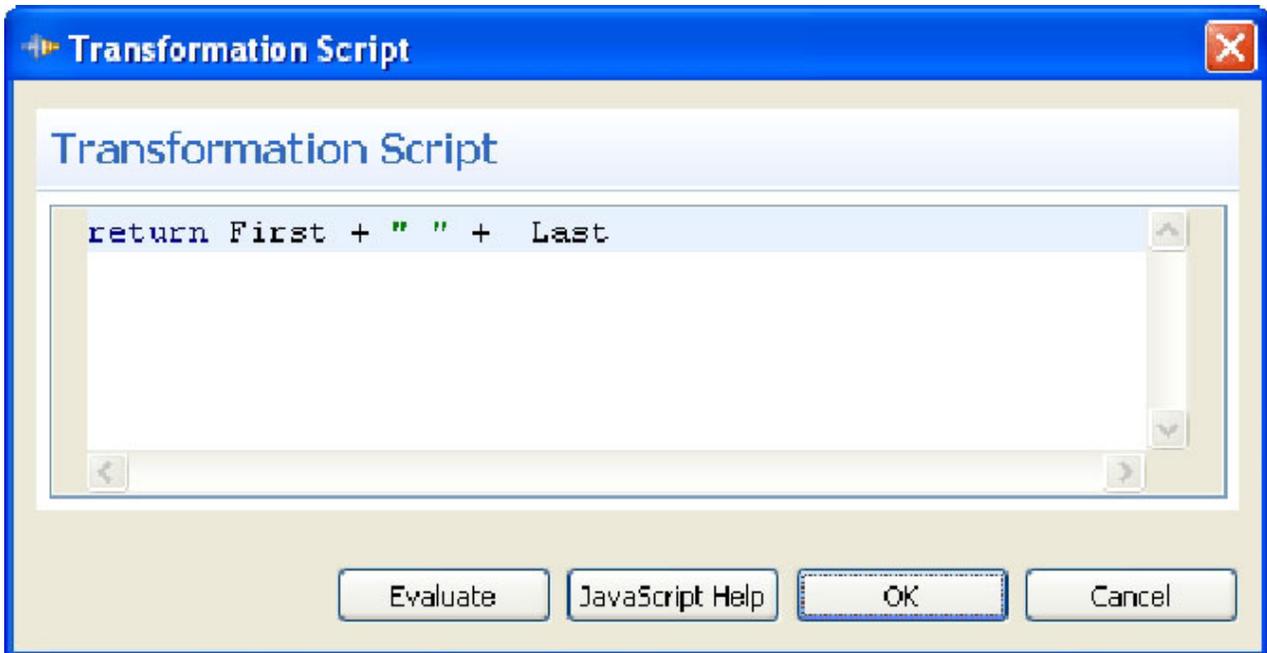


Figura 99. Mostrar script de transformación

Hay un botón **Evaluar** en el diálogo del editor Script de transformación para probar el script, y otro botón para generar consejos y ejemplos de JavaScript.

Pulse **Evaluar** ahora para hacerse una idea de cómo funciona la transformación.

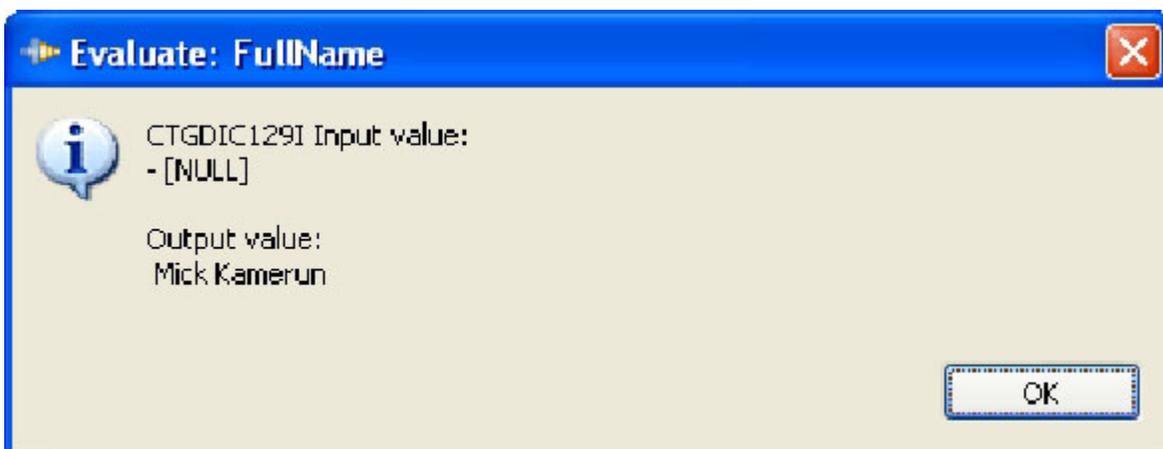
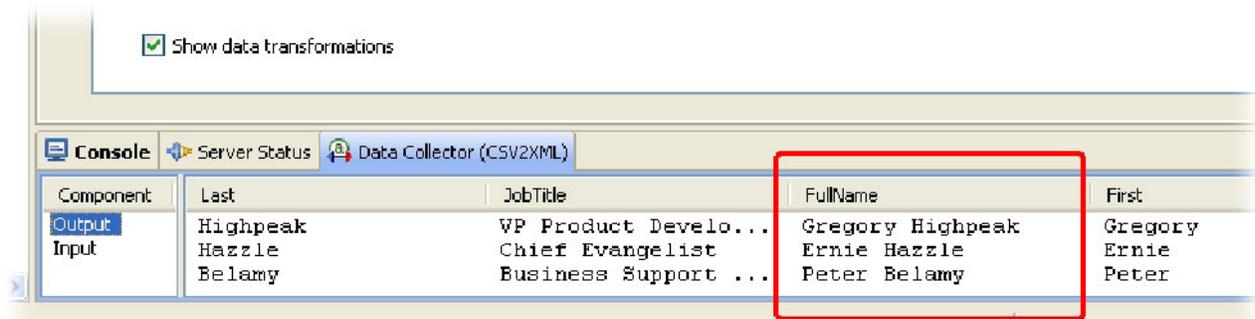


Figura 100. Evaluar expresión

El valor de salida mostrado se ha calculado utilizando los datos que ha recopilado al ejecutar la línea de ensamblaje. Cierre el diálogo de resultados de la evaluación pulsando **Aceptar**.

Después, pulse **Aceptar** para aceptar este script y cerrar el editor Script de transformación y volver a ejecutar la línea de ensamblaje EasyETL pulsando **Ejecutar** y luego visualizando las entradas recopiladas. Observe cómo ahora el **recopilador de datos** le ofrece dos recopilaciones de componente entre las que elegir: *salida* y *entrada*. Seleccione **Salida** y vea cómo el script de transformación ha

generado un valor 'FullName' para cada entrada.



The screenshot shows the 'Data Collector (CSV2XML)' console window. At the top, there is a checkbox labeled 'Show data transformations' which is checked. Below this, there are tabs for 'Console', 'Server Status', and 'Data Collector (CSV2XML)'. The main area displays a table with the following data:

Component	Last	JobTitle	FullName	First
Output	Highpeak	VP Product Develo...	Gregory Highpeak	Gregory
Input	Hazzle	Chief Evangelist	Ernie Hazzle	Ernie
	Belamy	Business Support ...	Peter Belamy	Peter

Figura 101. Recopilación de salida con el atributo FullName calculado

Así pues, ahora ya sabe cómo configurar un origen de entrada y seleccionar los atributos que se van a extraer, así como la manera de transformar estos datos para adaptarlos a sus necesidades de salida. El paso siguiente es seleccionar un destino de salida y llevar ahí los datos.

Selección de un destino de salida

Seleccione el **Conector de archivos** de su destino de salida usando el menú desplegable que está arriba de la vista de salida y guárdelo en un archivo denominado 'Output.xml'⁴⁵. Seleccione **Analizador XML** y pulse **Aceptar**. Tenga en cuenta que puede utilizar el botón **Conectar** para asegurarse de que la vía de acceso de archivo que ha introducido es válida. Sin embargo, no habrá datos que descubrir a menos que apunte el conector hacia un archivo XML existente.

Una vez configurada la salida, vuelva a ejecutar la línea de ensamblaje ETL. Cuando esté completada podrá abrir el archivo de salida y verificar los resultados.

45. Si no introduce una ruta completa o una relativa, entonces IBM Security Directory Integrator se basará en el Directorio de soluciones que especificó durante la instalación.

```

- <DocRoot>
  - <Entry>
    <First>Bill</First>
    <Last>Sanderman</Last>
    <Title>Chief Scientist</Title>
  </Entry>
  - <Entry>
    <First>Mick</First>
    <Last>Kamerun</Last>
    <Title>CEO</Title>
  </Entry>
  - <Entry>
    <First>Jill</First>
    <Last>Vox</Last>
    <Title>CTO</Title>
  </Entry>
  - <Entry>
    <First>Roger</First>
  </Entry>
  - <Entry>
    <First>Gregory</First>
    <Last>Highpeak</Last>
    <Title>VP Product Development</Title>
  </Entry>
  - <Entry>
    <First>Ernie</First>
    <Last>Hazzle</Last>
    <Title>Chief Evangelist</Title>
  </Entry>
  - <Entry>
    <First>Peter</First>
    <Last>Belamy</Last>
    <Title>Business Support Manager</Title>
  </Entry>
</DocRoot>

```

Figura 102. Salida XML

El destino de salida podría haber sido fácilmente una tabla de base de datos, del mismo modo que la entrada podría proceder de una aplicación Lotus Notes o de un directorio LDAP. Los pasos que ha realizado para crear este trabajo ETL de

ejemplo son los mismos, independientemente de los sistemas o almacenes de datos con los que trabaja.

Detección de cambios

IBM Security Directory Integrator proporciona varias características para detectar cambios en datos de entrada. Además de ofrecer un conjunto de conectores de detección de cambios, el usuario puede también habilitar el *motor delta* del origen de entrada.

El motor delta toma instantáneas de datos a medida que se leen y después las compara con las instantáneas tomadas durante la ejecución anterior para determinar qué ha cambiado. Las entradas no modificadas se omiten, y sólo las entradas modificadas se recuperan para su proceso en la línea de ensamblaje EasyETL.

Pulse el botón **Configurar** del origen de entrada y después seleccione la pestaña **Delta**.

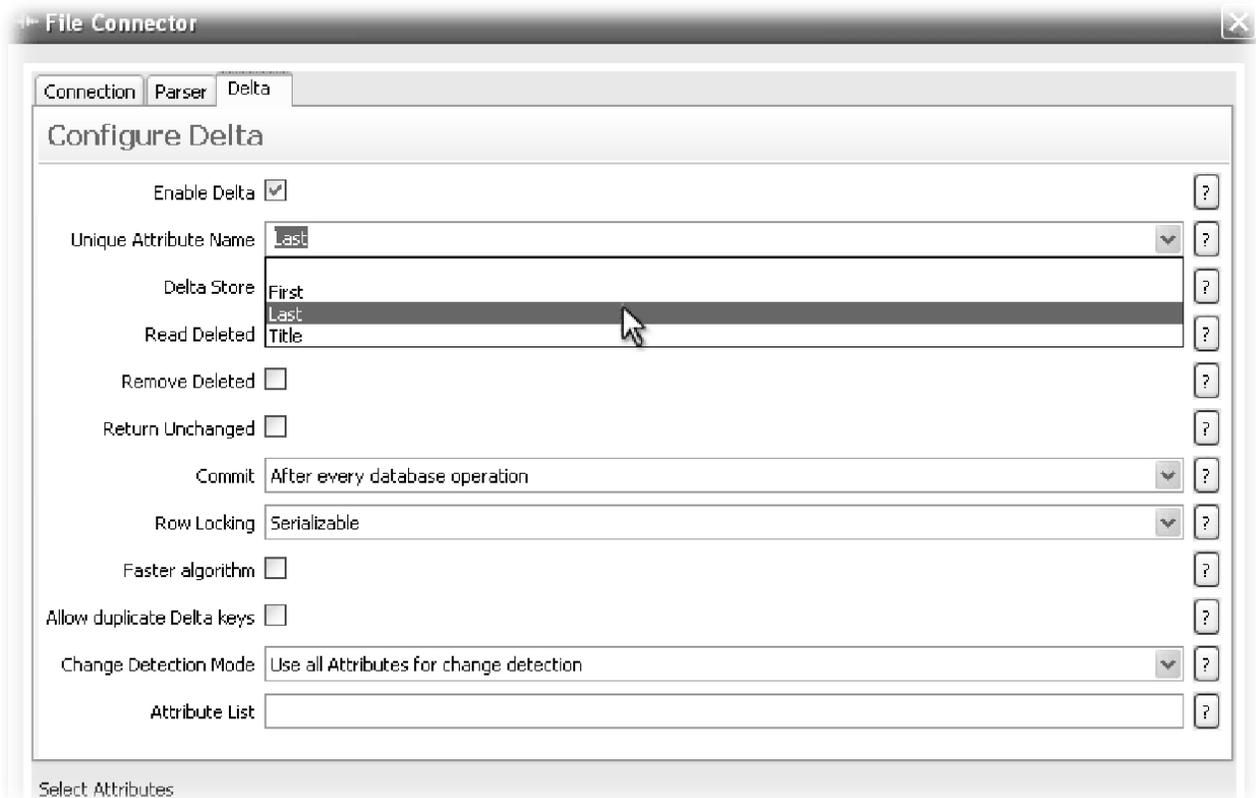


Figura 103. Configuración de delta

Primero debe habilitar el motor delta seleccionando la casilla de verificación que se encuentra en la parte superior del panel de configuración. Después, utilice el desplegable para seleccionar 'First' como **Nombre de atributo exclusivo**⁴⁶.

46. Como quizás haya deducido, el motor delta utiliza uno de los atributos de entrada para identificar de forma exclusiva las instantáneas. Si no hay ningún valor exclusivo disponible en los datos de entrada, puede especificar varios atributos que se concatenarán hacia y desde el ID de instantánea. Para hacerlo, escriba los nombres de varios atributos separados por un símbolo de adición (+). Por ejemplo: First + Last

Existen otros parámetros disponibles aquí, algunos de los cuales son más útiles cuando se trabaja en el entorno de trabajo estándar de IBM Security Directory Integrator y no en EasyETL. Por ejemplo, aunque una línea de ensamblaje EasyETL pueda detectar y transferir entradas nuevas y modificadas, no podrá suprimir una fila de una base de datos ni una entrada de un directorio. Sin embargo, escribirá esta información en un destino de salida, como un conector del sistema de archivos, con el analizador LDIF. Los archivos LDIF pueden contener etiquetas de operaciones de cambio, y algunos sistemas admiten la importación de LDIF.

En la siguiente página web encontrará más información sobre las características completas de gestión de deltas de IBM Security Directory Integrator:

http://www.tdi-users.org/twiki/pub/Integrator/HowTo/HowTo_SyncData_6.1.1070523.pdf

Es probable que desee realizar un cambio en el parámetro **Confirmar**. Este parámetro controla cuándo se confirman instantáneas nuevas y cambiadas en la base de datos del almacén del sistema de IBM Security Directory Integrator. De forma predeterminada, se establece en 'Después de cada operación de base de datos' y ocurre durante la fase de lectura.

Sin embargo, si desea asegurarse de que un cambio se ha transferido correctamente antes de confirmar la instantánea, establezca este desplegable en 'Al final del ciclo de la LE' para que la transferencia se lleve a cabo después de que el destino de salida se haya actualizado.

Para que el motor delta realice su trabajo, necesita un conjunto de instantáneas de línea base. Este se crea ejecutando el trabajo ETL por primera vez después de habilitar el delta. Una vez completado, observará que la ventana emergente notifica la escritura por duplicado. Esto se debe a que IBM Security Directory Integrator también contabiliza las instantáneas que se escriben en el almacén del sistema, por lo que resultan dos escrituras por cada entrada procesada.

Intente ejecutar de nuevo la línea de ensamblaje EasyETL y verá que esta vez no se escribe ninguna entrada. El motor delta ha detectado que los registros de entrada no se han modificado y los ha omitido.

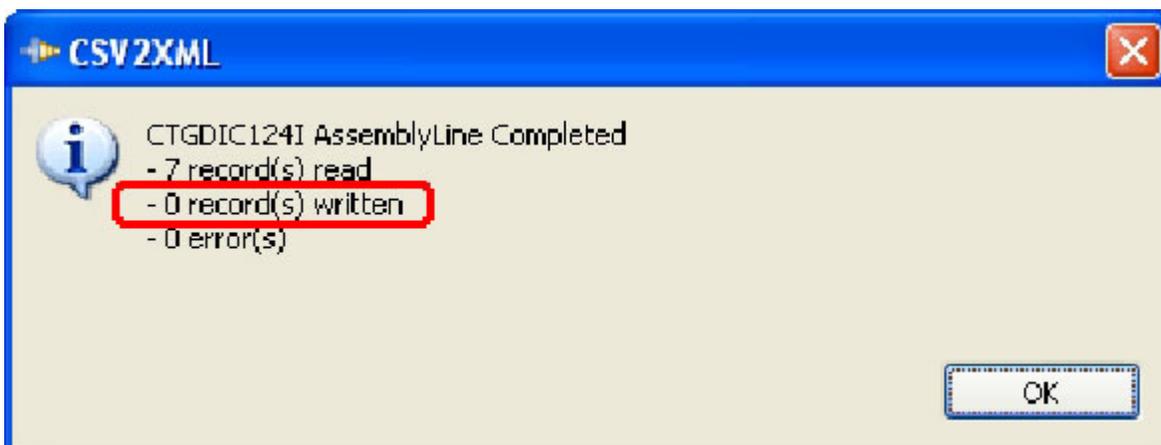


Figura 104. Todas las entradas sin modificar y omitidas

Como prueba final, abra el archivo CSV de entrada y cambie alguno de los valores de campo, excepto 'Last'⁴⁷. Guarde el cambio y después vuelva a ejecutar el trabajo ETL; verá que sólo se procesan las entradas modificadas.

Configuración del destino de salida para las actualizaciones

La configuración actual funciona bien para la salida en un archivo. No obstante, si guardara estos cambios en un directorio, RDBMS o almacén de datos similar, desearía añadir datos nuevos así como actualizar los registros existentes. Para que el trabajo EasyETL haga esto, primero debe seleccionar qué atributo de salida desea utilizar como criterio para localizar el registro que se va a modificar.

Esto se lleva a cabo pulsando el botón derecho del ratón en el atributo de salida deseado y seleccionando la opción **Utilizar como criterio de enlace** option.

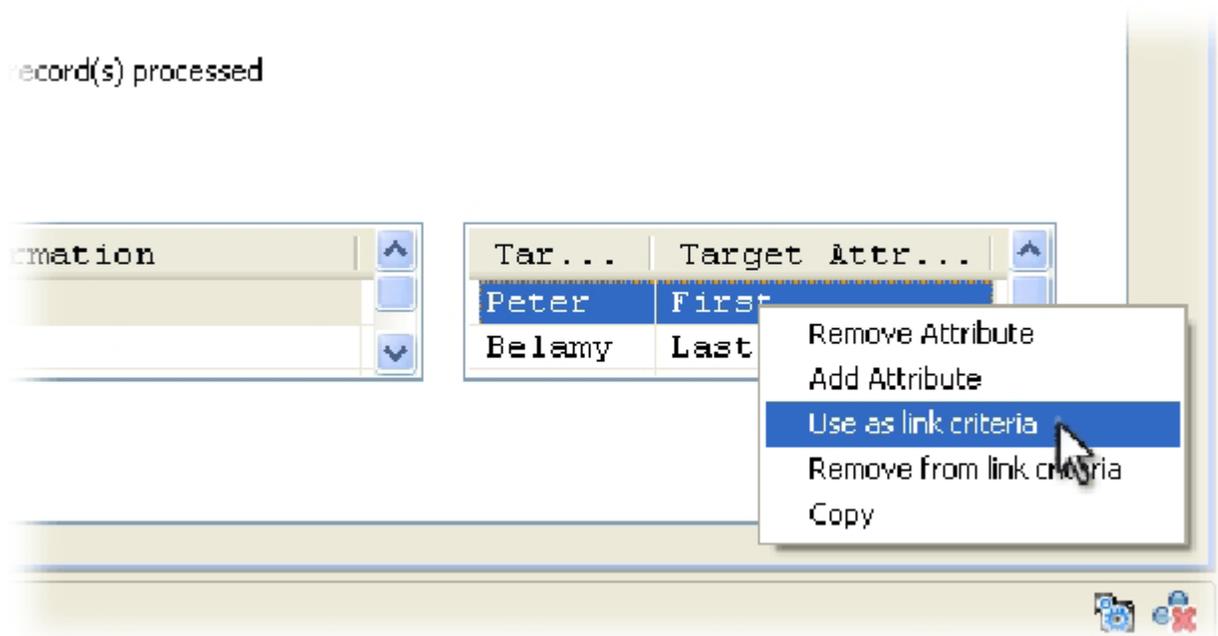


Figura 105. Selección del criterio de enlace

Cuando el conector de salida escribe en el destino, primero busca un registro utilizando el atributo Criterio de enlace especificado. Si no encuentra ninguna coincidencia, se añade una nueva entrada. Si se encuentra una coincidencia, este registro se actualiza.

Es así de simple: el trabajo ETL se ha configurado ahora para proporcionar sincronización en curso entre el origen de entrada y el destino de salida.

Activos de línea de mandatos para ejecutar y planificar el trabajo ETL

Una vez que la línea de ensamblaje ETL esté preparada para el despliegue, puede pulsar el botón derecho del ratón en el proyecto del navegador y seleccionar la

47. Dado que este es el atributo utilizado para identificar instantáneas, el cambio efectuado en el valor de una entrada hará que aparezca como un *nuevo* registro en el motor delta

opción **Crear archivos necesarios...**

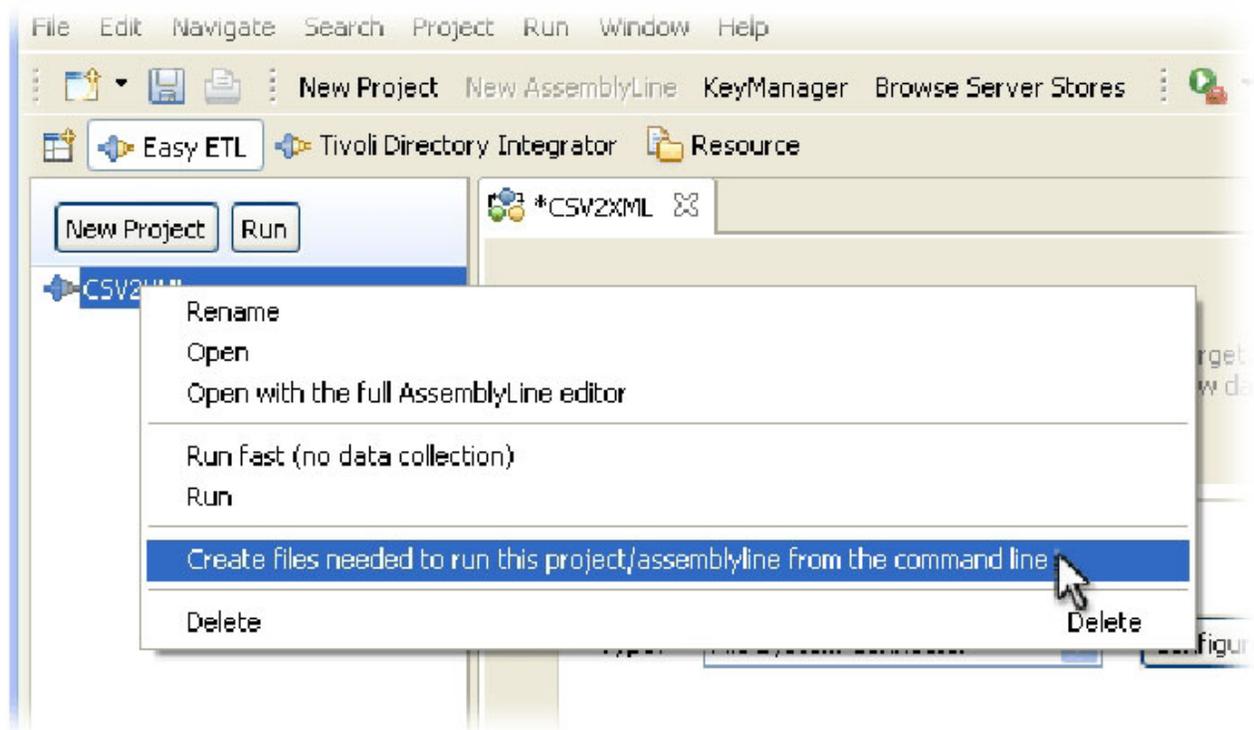


Figura 106. Creación de activos de línea de mandatos para ejecutar el trabajo ETL

Hace que aparezca el diálogo Exportar archivos en el que escribir este archivo de proceso por lotes/script. Tenga en cuenta que se le asignará el mismo nombre que al proyecto, así que si este ejercicio de guía de aprendizaje se ejecuta en Windows, se llamará 'CSV2XML.bat'. La ejecución del proyecto EasyETL desde la línea de mandatos proporciona el rendimiento máximo a la solución.

También obtendrá un archivo XML creado en la misma ubicación. Este archivo es el archivo *Config* de IBM Security Directory Integrator y contiene los detalles de la línea de ensamblaje EasyETL que el servidor de IBM Security Directory Integrator necesita para ejecutarla. Si abre el script generado en un editor de texto, verá la línea necesaria para iniciar un servidor de IBM Security Directory Integrator; haga que la línea apunte a un archivo Config y especifique la línea de ensamblaje que se debe ejecutar. Todo lo que debe hacer ahora es configurar una tarea planificada o trabajo cronológico para que invoque periódicamente este script y el servicio de sincronización/migración quedará establecido.

Opciones adicionales

ETL de alta velocidad

Aunque el recopilador de datos es una herramienta poderosa, la línea de ensamblaje ETL se ejecuta más despacio debido a la recopilación de datos y su presentación en la pantalla. Si por el contrario desea que la línea de ensamblaje EasyETL se procese lo más rápidamente posible, puede seleccionar el proyecto y pulsar el botón Ejecutar, situado en la parte superior del navegador, o pulsar el botón derecho del ratón en el proyecto y seleccionar la opción **Ejecución rápida...**

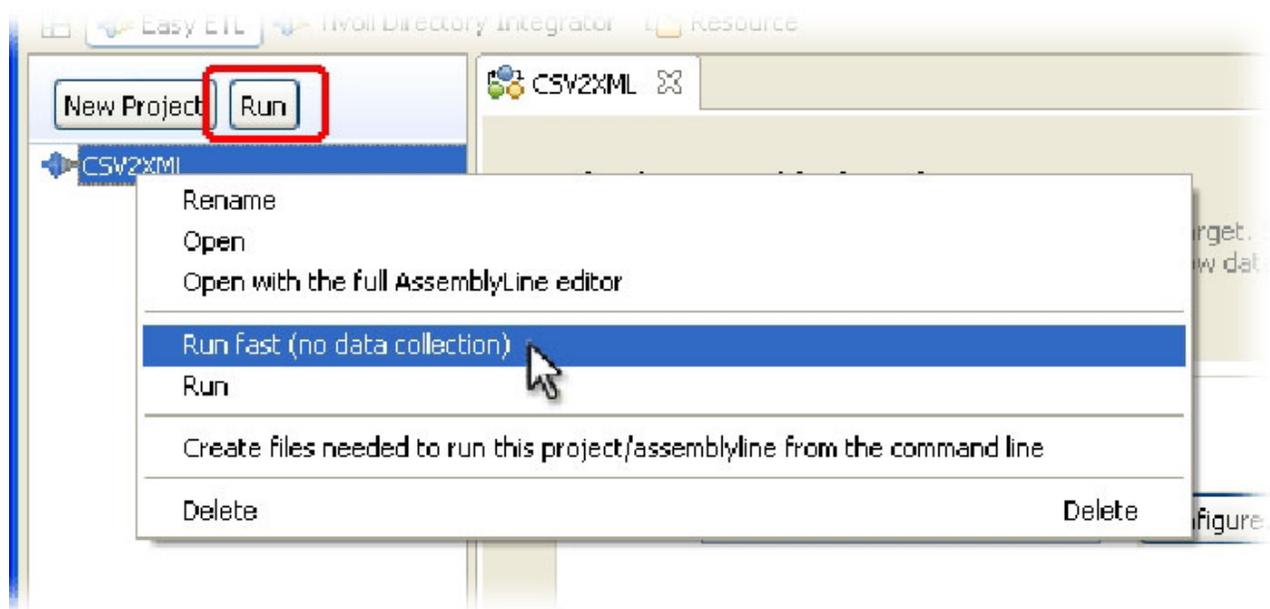


Figura 107. Ejecución del trabajo ETL a la máxima velocidad

Cualquiera de las opciones abrirá una pantalla de consola en la que aparecerán mensajes de registro de la línea de ensamblaje mientras la línea de ensamblaje se ejecuta a la máxima velocidad.

Tenga en cuenta que la opción **Ejecutar** del menú contextual de proyecto ejecuta el trabajo ETL con la recopilación de datos.

Filtrado del conjunto de datos de entrada

Otra poderosa característica es la capacidad de controlar el contenido del conjunto de datos de entrada. Esta característica está disponible siempre que el origen de entrada es una base de datos o un directorio.

Por ejemplo, seleccione 'Conector LDAP' para la entrada y eche un vistazo al diálogo de configuración de este componente. Junto al parámetro **Filtro de búsqueda** hay un botón etiquetado con tres puntos (...). Este botón abre el editor Criterio de enlace, donde se pueden definir reglas de búsqueda que se aplicarán a crear el conjunto de resultados que este conector leerá.

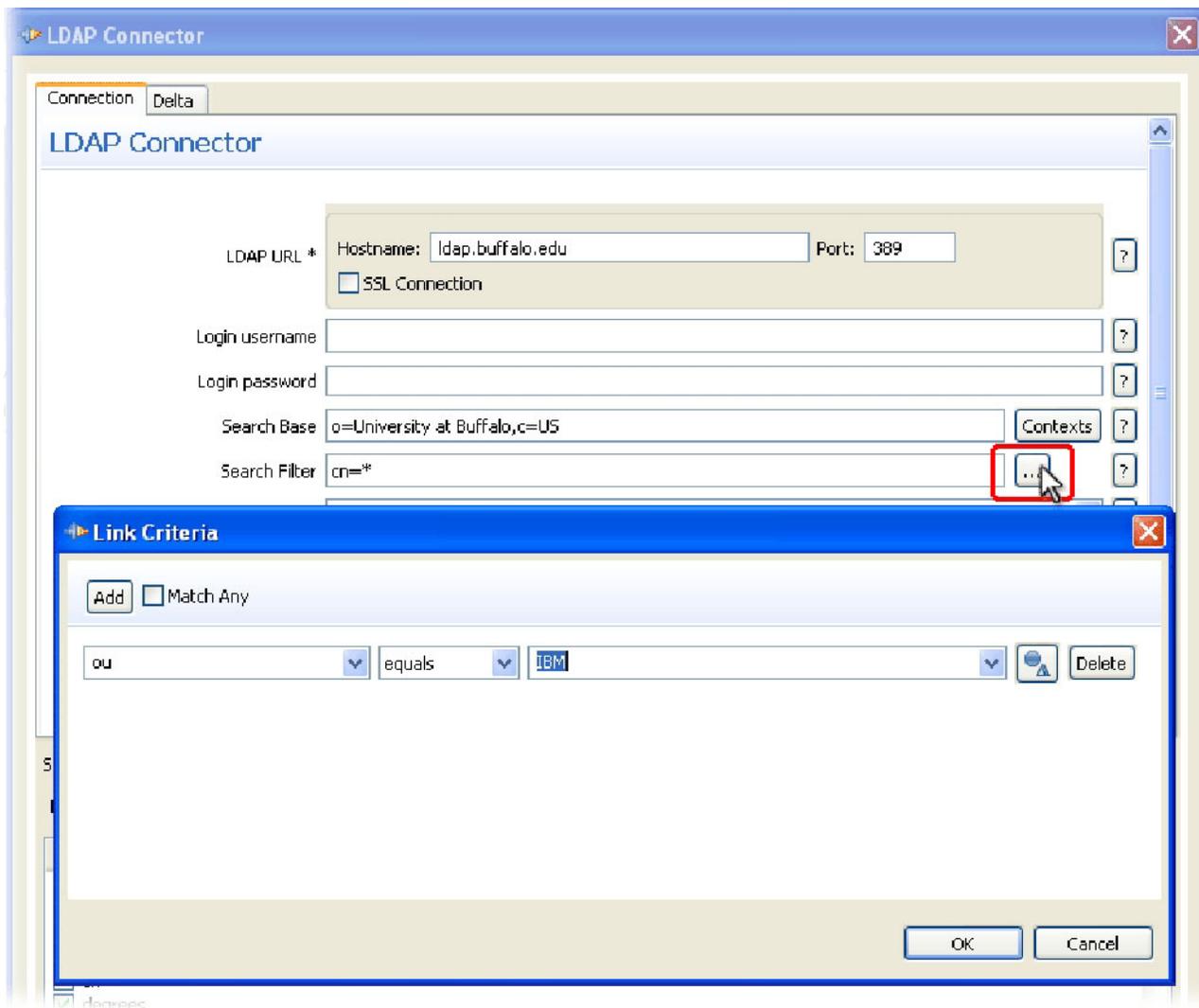


Figura 108. Definición de criterios de enlace para un conector de entrada

Esta misma característica está disponible para los conectores de base de datos y JDBC, donde encontrará el botón (...) junto al parámetro **Seleccionar**.

Aunque puede introducir usted mismo la sintaxis de búsqueda LDAP directamente en el parámetro de búsqueda, esto requiere que conozca la sintaxis de los filtros de búsqueda LDAP o las sentencias de selección JDBC. Suele ser más fácil expresar la selección deseada utilizando el criterio de enlace y dejando que el conector maneje la sintaxis subyacente.

Cómo llevar la línea de ensamblaje EasyETL al siguiente nivel

Abrir el proyecto ETL en el editor de línea de ensamblaje IBM Security Directory Integrator con funciones completas le permite añadir registros personalizados y auditoría, gestión de errores, lógica de migración tras error, reconexión automática, aumento de datos (uniones) y mucho más a la solución de migración o sincronización. Para ello, pulse el botón derecho del ratón en un proyecto y seleccione la opción **Abrir con editor de línea**

de ensamblaje completo. Todavía trabajará en el entorno de trabajo EasyETL, pero podrá acceder a funciones adicionales disponibles en la línea de ensamblaje.

Si esto es de su agrado y está preparado para el cambio, cambie a la perspectiva de Security Directory Integrator (**Windows > Abrir perspectiva > Security Directory Integrator**) y empiece a trabajar en el entorno de trabajo completo de IBM Security Directory Integrator. O mejor todavía, ahora que domina el manejo de EasyETL, vuelva a la sección 1 y comience a indagar las prestaciones completas de IBM Security Directory Integrator.

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en EE.UU. Es posible que IBM no ofrezca en otros países los productos, servicios o características que se describen en este documento. Consulte al representante local de IBM para obtener información acerca de los productos y servicios que actualmente están disponibles en su localidad. Las referencias a productos, programas o servicios IBM no pretenden afirmar ni dar a entender que únicamente puedan utilizarse dichos productos, programas o servicios IBM. En su lugar, puede utilizarse cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. La entrega de este documento no le otorga ninguna licencia sobre esas patentes. Puede enviar consultas sobre licencias, por escrito a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

Para consultas sobre licencias en las que se solicite información sobre el juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de Propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japón

El párrafo siguiente no se aplica en el Reino Unido ni en ningún otro país donde estas disposiciones no sean coherentes con la legislación local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O ADECUACIÓN PARA UN FIN DETERMINADO.

Algunos estados no permiten la renuncia a garantías explícitas o implícitas en determinadas transacciones. Por lo tanto, es posible que estas indicaciones no apliquen en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede realizar en cualquier momento mejoras o cambios en los productos o programas descritos en esta publicación sin previo aviso.

Cualquier referencia hecha en esta información a sitios Web que no son de IBM se proporciona únicamente para su comodidad y no debe considerarse en modo alguno como una aprobación de esos sitios Web. La información de esos sitios web no forma parte de la información del presente producto de IBM y la utilización de esos sitios web se realiza bajo la propia responsabilidad del usuario.

IBM puede utilizar o distribuir la información que se le proporcione del modo que estime apropiado sin incurrir por ello en ninguna obligación con el remitente.

Los titulares de licencias de este programa que deseen información sobre el mismo con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido éste) y (ii) la utilización mutua de la información intercambiada, deben ponerse en contacto con:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Esta información puede estar disponible, sujeta a las condiciones y los términos apropiados, incluyendo en ciertos casos el pago de una cuota.

El programa bajo licencia descrito en esta información y todo el material bajo licencia disponible para el mismo los proporciona IBM bajo los términos de las Condiciones Generales de IBM, el Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre IBM y el cliente.

Cualquier dato sobre rendimiento que contiene este documento se ha determinado en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar de forma significativa. Es posible que se hayan realizado algunas mediciones en sistemas a nivel de desarrollo y no existe ninguna garantía que dichas mediciones sean iguales en los sistemas de disponibilidad general. Además, algunas de las mediciones pueden haberse estimado mediante extrapolaciones. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relacionada con productos que no son de IBM procede de los proveedores de dichos productos, sus anuncios publicados u otras fuentes de disponibilidad general. IBM no ha comprobado estos productos y no puede confirmar la precisión del rendimiento, compatibilidad o cualquier otra afirmación relacionada con productos que no son de IBM. Las consultas sobre las posibilidades de los productos que no son de IBM se deben dirigir a los proveedores de dichos productos.

Todas las declaraciones relativas a las intenciones futuras de IBM están sujetas a cambios o cancelaciones sin previo aviso y únicamente representan planes y objetivos.

Todos los precios de IBM que se muestran son precios de IBM sugeridos para comercios minoristas, están actualizados y están sujetos a posibles cambios sin previo aviso. Los precios de cada comercio específico pueden variar.

Esta información sirve únicamente a modo de planificación. La información contenida puede cambiar antes de que los productos descritos se encuentren a disposición del público.

Esta información contiene ejemplos de datos e informes empleados en operaciones empresariales diarias. A fin de ofrecer una ilustración lo más completa posible, los ejemplos incluyen los nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con los nombres y las direcciones empleados por una empresa real es totalmente casual.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente, que ilustran las técnicas de programación en diferentes plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier modo sin abonar nada a IBM, para fines de desarrollo, uso, comercialización o distribución de programas de aplicación que se ajusten a la interfaz del programa de aplicación de la plataforma operativa para la que se han escrito estos programas de ejemplo. Estos ejemplos no se han comprobado detenidamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni implicar la fiabilidad, la capacidad de servicio ni el funcionamiento de estos programas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier modo, sin pago alguno a IBM, con fines de desarrollo, uso, comercialización o distribución de programas de aplicación que se ajusten a las interfaces de programa de aplicación de IBM.

Cada copia o parte de estos programas de ejemplo o trabajo derivado del mismo, debe incluir un aviso de copyright como el que se indica a continuación:

© (nombre de su empresa) (año). Partes de este código se derivan de programas de ejemplo de IBM Corp. © Copyright IBM Corp. _escriba el año o años_. Reservados todos los derechos.

Si está viendo esta información en un formato legible por máquina, es posible que no aparezcan las fotografías y las ilustraciones.

Marcas registradas

IBM, el logotipo de IBM e ibm.com son marcas registradas de International Business Machines Corp., registradas en numerosas jurisdicciones de todo el mundo. Otros nombres de servicios y productos pueden ser marcas registradas de IBM u otras empresas. Encontrará una lista actualizada de marcas registradas de IBM en la sección "Información sobre copyright y marcas registradas" del sitio web www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript y todas las marcas registradas basadas en Adobe son marcas registradas de Adobe Systems Incorporated en EE.UU. o en otros países.

IT Infrastructure Library es una marca registrada de la CCTA (Central Computer and Telecommunications Agency) que actualmente forma parte de la OGC (Office of Government Commerce).

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas registradas de Intel Corporation o de sus empresas filiales en los Estados Unidos o en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos o en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos o en otros países.

ITIL es una marca registrada y una marca registrada comunitaria de OGC (Office of Government Commerce), y está registrada en la Oficina de Patentes y Marcas de Estados Unidos.

UNIX es una marca registrada de Open Group en los Estados Unidos y en otros países.



Java y todas las marcas registradas y logotipos basados en Java son marcas registradas de Oracle o de sus empresas filiales.

Cell Broadband Engine es una marca comercial de Sony Computer Entertainment, Inc. en los Estados Unidos o en otros países y se utiliza bajo licencia.

Linear Tape-Open, LTO, el logotipo de LTO, Ultrium y el logotipo de Ultrium son marcas registradas de HP, IBM Corp. y Quantum en Estados Unidos y en otros países.

Índice

A

accesibilidad ix
AMC 71
ampliación 86
añadir 8
append 83
archivos de la guía de aprendizaje 1
Atributo 5

B

buscar 8, 66
búsqueda de datos 48

C

coincidencia 66
Cómo empezar 8
Comportamiento ante un valor nulo 32
conectividad 83
Conexión 6
conexión perdida 83
consola 72
Correlación de atributos 6, 15
Correlación de entrada 6
Correlación de salida 6
Criterio de enlace 66
cursos ix

D

Datos 5
datos que faltan 32
Depuración 40
depurador 87
desencadenante 71
determinación de problemas ix
diseño de soluciones 1, 3
disponibilidad 81, 84

E

Ejecutar 30
empezar soluciones muy básicas 3
Entrada 5
error de inicialización 83
errores de ejecución 67
excepciones 84

F

flujo de datos 6
formación ix
fortalecimiento 81

H

herencia 63

I

IBM
 Soporte de software ix
 Support Assistant ix
IBM Security Directory Integrator,
 proyectos 1
iterar 8

J

JavaScript 1

L

legibilidad 81
línea de ensamblaje 6, 15, 30
línea de mandatos 71
log4j 83

M

manejo de sucesos 73
mantenimiento 81
Modalidad Buscar 59, 66

P

planificación 72
planificador 72

R

registro cronológico 83
rendimiento 86
repetidor de pasos 40
resolución de problemas ix
responder 71
reutilización 81

S

salida del registro 30
seguridad 81
servidor web 73
soluciones utilizando IBM Security
 Directory Integrator 3
Suceso 71
Sucesos 71

T

TINA 73
tolerancia a fallos 81

U

unir 48

V

Valor 5
Valores nulos 32



Impreso en España

GI11-8676-03

