

IBM Security Directory Integrator
Versão 7.2.0.1

Guia de Introdução



IBM Security Directory Integrator
Versão 7.2.0.1

Guia de Introdução



Nota

Antes de usar estas informações e o produto suportado por elas, leia as informações gerais no “Avisos” na página 111.

Aviso da Edição

Nota: Esta edição se aplica à versão 7.2.0.1 do *IBM Security Directory Integrator* e a todas as liberações e modificações subsequentes, até que seja indicado de outra maneira em novas edições.

© Copyright IBM Corporation 2003, 2014.

Índice

Figuras	v	Decifrando Erros de Execução	67
Sobre esta Publicação	vii	Capítulo 3. Integração Acionada por Eventos	71
Acesso a Publicações e Terminologia	vii	Planejando AssemblyLines	72
Acessibilidade	ix	AssemblyLines de Pedido de Serviço	73
Treinamento Técnico	ix	Capítulo 4. Fortalecendo suas Soluções de Integração	81
Informações de Suporte	ix	Legibilidade, Reutilização e Capacidade de Configuração	81
Declaração das Boas Práticas de Segurança	ix	Criação de Log e Auditoria	83
Capítulo 1. Introdução.	1	Problemas de Conectividade.	83
Simplificar e solucionar.	3	Disponibilidade do AssemblyLine	84
Arquitetura do Kernel/Componente	4	Escala e Desempenho	86
Modelo de Dados de Entrada-Atributo-Valor	5	Monitorando	87
Fluxos de Dados = AssemblyLines	6	O Depurador de AssemblyLine.	87
Introdução	8	Apêndice. Guia do EasyETL	89
Capítulo 2. Introdução ao IBM Security Directory Integrator	11	Criando um Projeto.	92
Criando seu Primeiro AssemblyLine	15	Detectando Mudanças	103
Executando seu Assemblyline	29	Avisos	111
Comportamento Nulo: Lidando com Atributos/Valores Ausentes	32	Índice Remissivo	115
Depurando seu AssemblyLine	40		
Consultando Dados de uma Origem Sequencial	48		
Usando o Modo de Consulta	59		
Herança	63		
Consultar Regras de Procura = Critérios de Link	66		

Figuras

1. Um Modelo de Dados de Entrada-Atributo-Valor	6	41. Depurando seu AssemblyLine	40
2. Dados Fluindo em um AssemblyLine	7	42. O Data Stepper do AssemblyLine	41
3. Cenário do Tutorial	8	43. Avançando na Execução do AL	42
4. Iniciando o Configuration Editor	11	44. Avançando para o Conector Write_XML_File	43
5. Selecionando sua Área de Trabalho.	11	45. Modo Depurador Avançado	44
6. Tela de Boas-vindas do Configuration Editor	12	46. Botões do Depurador	45
7. Nomeando seu Novo Projeto.	13	47. Configurando um Ponto de Interrupção	46
8. Tela Principal do Config Editor	14	48. Configurando um Ponto de Interrupção no Script	47
9. Diagrama de Cenário Simplificado com Somente Duas Origens de Dados	15	49. Linha de Comandos de Avaliação de JavaScript	48
10. Caixa de Diálogo Novo AssemblyLine	16	50. O Fluxograma do Cenário.	49
11. Editor de AssemblyLine Vazio	17	51. Arrastando 'FullName' para o Mapa de Entrada de seu Conector de Agente Iterativo	49
12. Inserindo um Novo Componente	18	52. Editando a Designação para 'FullName'	50
13. Escolhendo o Componente	19	53. Arrastar o ConnectorLoop	51
14. Renomeando o Conector e Alterando seu Modo	20	54. Configuração de ConnectorLoop	51
15. Painel de Configuração do Conector de Arquivo.	21	55. Configurações Avançadas de ConnectorLoop	52
16. Selecionando o Analisador Durante Inserir Novo Objeto	22	56. Atributos Hierárquicos.	52
17. Seleção do Menu de Contexto Pesquisar Dados	23	57. Arrastando do Esquema para o Mapa de Atributos	53
18. O Navegador de Dados	24	58. Editor de Condição para a Ramificação IF	53
19. Descobrimo o Esquema Interativamente pela Pesquisa de Dados Ativos.	24	59. Script do Gancho Fim dos Dados	54
20. AL com Conector de Agente Iterativo no Local	25	60. Lista de Componentes na seção Fluxo de Dados do AssemblyLine	55
21. Botão Incluir Componente	26	61. Script de uma Condição para a Ramificação IF	56
22. AL com Dois Conectores no Local	27	62. AssemblyLine Completado com Loop FOR-EACH	57
23. Arrastando Atributos para o Mapa de Saída	28	63. Saída do Log com Estatísticas da Ramificação IF.	58
24. Renomeando uma Regra do Mapa de Atributos	28	64. Saída XML com Atributo 'telephoneNo'	59
25. Incluindo o Atributo 'FullName' no Mapa de Saída.	29	65. Função Copiar do AssemblyLine	60
26. Editando a Designação.	29	66. Copiando um AssemblyLine para seu Projeto	61
27. O Botão Executar	30	67. Executar o AL CreatePhoneDB	61
28. Saída do Log a partir da Execução do AssemblyLine.	30	68. Saída do Log a partir do AssemblyLine 'CreatePhoneDB'	62
29. Barra de Botões para a Janela Saída do Log	30	69. Arrastar um Conector para Recursos	63
30. Pesquisando Dados Criados por um Conector de Saída	31	70. Arrastar o Novo Recurso para seu AssemblyLine.	63
31. Pesquisando o XML Resultante	32	71. Configurando a Herança para a Guia Ganchos	64
32. Botão Comportamento Nulo para Configuração no Nível do AL	33	72. Restaurando a Herança para uma Regra de Mapeamento	65
33. Diálogo de Configuração do Comportamento Nulo.	34	73. Alterando o Modo, Descobrimo e Mapeando Atributos	65
34. Resultado na Saída XML de Configurações do Comportamento Nulo	34	74. Um Critério de Link Simples.	66
35. Selecionando o Componente de Ramificação IF	35	75. Mensagem de Erro na Saída do Log	67
36. Editando Condições para a Ramificação IF	36	76. Fluxograma Parcial para o Modo de Consulta	68
37. Incluindo Condições Simples para a Ramificação IF	37	77. Primeiro Exercício do Tutorial Concluído	69
38. Seu Primeiro AssemblyLine Concluído	38	78. Planejador do IBM Security Directory Integrator	73
39. Reconfigurando o Comportamento Nulo para o AssemblyLine	39	79. Painel Mapa de Atributos do Conector de Servidor HTTP	74
40. Saída do Log com suas Mensagens e o Dump de Entrada de Trabalho	39	80. Incluir Item do Mapa de Atributos de Entrada	75
		81. Item do Mapa de Curingas	76
		82. Propriedades de Cabeçalho TCP e HTTP Retornadas como Atributos	77

83. Arrastar no AssemblyLine Function Component (AL FC)	78	97. EasyETL AssemblyLine Concluído	98
84. Dump de Entrada de Trabalho Seguido por Estatísticas do AL	79	98. Ativando Transformação	99
85. TINA_WebServer AssemblyLine Concluído	80	99. Mostrar Script de Transformação	100
86. Interface da Web Simples para sua Solução	80	100. Avaliar Expressão	100
87. Tela de Boas-Vindas.	90	101. Coleta de Saída com o Atributo FullName Calculado.	101
88. Figura 2. Ambiente de Trabalho do EasyETL	91	102. Saída XML	102
89. Botão Novo Projeto	92	103. Configuração Delta	103
90. Editor Simple AssemblyLine	92	104. Todas as Entradas Inalteradas e Ignoradas	104
91. Selecionando Informações de Origem	93	105. Selecionando os Critérios de Link.	105
92. Configurando o Parâmetro Caminho de Arquivo.	94	106. Criando recursos da linha de comandos para executar a tarefa ETL	106
93. Testando a Conexão e Descobrimdo o Esquema	95	107. Executando a Tarefa ETL em Velocidade Máxima	107
94. Origem de Entrada Configurada	96	108. Definindo Critérios de Link para um Conector de Entrada	108
95. Renomeando um Atributo de Saída	97		
96. Um Registro Lido e Coletado	98		

Sobre esta Publicação

Use o Guia de Introdução do IBM® Security Directory Integrator para aprender conceitos básicos do Security Directory Integrator e para entender o histórico sobre como projetar soluções efetivas de integração de dados.

Este documento apresenta informações conceituais sobre o IBM Security Directory Integrator e fornece exemplos para ajudar a iniciar o uso do produto.

Acesso a Publicações e Terminologia

Leia as descrições da biblioteca do IBM Security Directory Integrator Versão 7.2.0.1 e as publicações relacionadas que podem ser acessadas online.

Esta seção fornece:

- Uma lista de publicações na “Biblioteca do IBM Security Directory Integrator”.
- Links para “Publicações Online” na página viii.
- Um link para “Website IBM Terminology” na página ix.

Biblioteca do IBM Security Directory Integrator

Os documentos a seguir estão disponíveis na biblioteca do IBM Security Directory Integrator:

- *IBM Security Directory Integrator Versão 7.2.0.1 Federated Directory Server Administration Guide*

Contém informações sobre como usar o console do Federated Directory Server para projetar, implementar e administrar soluções de integração de dados. Também contém informações sobre como usar o protocolo System for Cross-Domain Identity Management (SCIM) e a interface para gerenciamento de identidade.

- *IBM Security Directory Integrator Versão 7.2.0.1 Getting Started Guide*

Contém um breve tutorial e uma introdução ao IBM Security Directory Integrator. Inclui exemplos para criar interação e aprendizado acessível do IBM Security Directory Integrator.

- *IBM Security Directory Integrator Versão 7.2.0.1 Users Guide*

Contém informações sobre como utilizar o IBM Security Directory Integrator. Contém instruções para projetar soluções utilizando a ferramenta de designer do Security Directory Integrator (o Editor de Configuração) ou para executar as soluções prontas da linha de comandos. Também fornece informações sobre interfaces, conceitos e criação da AssemblyLine.

- *IBM Security Directory Integrator Versão 7.2.0.1 Installation and Administrator Guide*

Inclui informações completas sobre instalação, migração a partir de uma versão anterior, configuração da funcionalidade de criação de log e o modelo de segurança subjacente da API do Servidor Remoto do IBM Security Directory Integrator. Contém informações sobre como implementar e gerenciar soluções.

- *IBM Security Directory Integrator Versão 7.2.0.1 Guia de Referência*

Contém informações detalhadas sobre os componentes individuais do IBM Security Directory Integrator: Conectores, Componentes de Função, Analisadores, Objetos, etc. - os blocos de construção da AssemblyLine.

- *IBM Security Directory Integrator Versão 7.2.0.1 Problem Determination Guide*
Fornece informações sobre as ferramentas, os recursos e as técnicas do IBM Security Directory Integrator que podem ajudar a identificar e resolver problemas.
- *IBM Security Directory Integrator Versão 7.2.0.1 Message Guide*
Fornece uma lista de todas as mensagens informativas, de aviso e de erro associadas ao IBM Security Directory Integrator.
- *IBM Security Directory Integrator Versão 7.2.0.1 Password Synchronization Plug-ins Guide*
Inclui informações completas para instalação e configuração de cada um dos cinco Plug-ins de Sincronização de Senha da IBM: Windows Password Synchronizer, Sun Directory Server Password Synchronizer, IBM Security Directory Server Password Synchronizer, Domino Password Synchronizer e Password Synchronizer para UNIX e Linux. Também fornece instruções de configuração para o LDAP Password Store e o JMS Password Store.
- *IBM Security Directory Integrator Versão 7.2.0.1 Release Notes*
Descreve os novos recursos e as informações mais recentes sobre o IBM Security Directory Integrator que não tenham sido incluídos na documentação.

Publicações Online

A IBM posta publicações do produto quando o produto é liberado e quando as publicações são atualizadas nos locais a seguir:

Biblioteca do IBM Security Directory Integrator

O site de documentação do produto (<http://www-01.ibm.com/support/knowledgecenter/SSCQGF/welcome>) exibe a página de boas-vindas e a navegação para essa biblioteca.

IBM Security Systems Documentation Central

O IBM Security Systems Documentation Central fornece uma lista alfabética de todas as bibliotecas do produto IBM Security System s e links para a documentação online para versões específicas de cada produto.

Centro de Publicações IBM

O site do Centro de Publicações IBM (<http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>) oferece funções de procura customizada para ajudar você a localizar todas as publicações IBM necessárias.

Informações Relacionadas

As informações relacionadas ao IBM Security Directory Integrator estão disponíveis nos locais a seguir:

- O IBM Security Directory Integrator usa o cliente JNDI da Oracle. Para obter informações sobre o cliente JNDI, consulte *Java Naming and Directory Interface™ Specification* em <http://download.oracle.com/javase/7/docs/technotes/guides/jndi/index.html> .
- Informações que podem ajudar a responder suas perguntas relacionadas aos IBM Security Directory Integrator podem ser localizadas em https://www-947.ibm.com/support/entry/myportal/over-accesspubsview/software/security_systems/tivoli_directory_integrator.

Website IBM Terminology

O website IBM Terminology consolida a terminologia para bibliotecas do produto em um local. É possível acessar o website de Terminologia em <http://www.ibm.com/software/globalization/terminology>.

Acessibilidade

Os recursos de acessibilidade ajudam usuários com alguma deficiência física, como mobilidade restrita ou visão limitada, a usarem produtos de software com êxito. Com este produto, você pode utilizar tecnologias de assistência para escutar e navegar na interface. Também é possível usar o teclado, em vez do mouse, para operar todos os recursos da interface gráfica com o usuário.

Para obter informações adicionais, consulte o Apêndice de Acessibilidade no *Configurando o Directory Integrator*.

Treinamento Técnico

Para obter informações sobre treinamento técnico, consulte o seguinte website do IBM Education em <http://www.ibm.com/software/tivoli/education>.

Informações de Suporte

O Suporte IBM fornece assistência para problemas e rotina relacionados a código, instalação de curta duração ou questões de uso. Você pode acessar diretamente o site Suporte de Software IBM em <http://www.ibm.com/software/support/probsub.html>.

O *Resolução de problemas* fornece detalhes sobre:

- Quais informações coletar antes de contatar o Suporte IBM.
 - Os vários métodos para contatar o Suporte IBM.
 - Como usar o IBM Support Assistant.
 - Instruções e recursos de determinação de problema para isolar e corrigir você mesmo o problema.
-

Declaração das Boas Práticas de Segurança

A segurança do sistema de TI envolve a proteção de sistemas e informações por meio de prevenção, detecção e resposta para acesso incorreto de dentro e fora da empresa. O acesso incorreto pode resultar em informações alteradas, destruídas, desapropriadas ou usurpadas, ou pode resultar em danos ou usurpação de seus sistemas, incluindo o uso em ataques a terceiros. Nenhum produto ou sistema de TI deve ser considerado completamente seguro e nenhum produto, serviço ou medida de segurança pode ser completamente efetivo na prevenção do acesso ou uso incorreto. Os sistemas, produtos e serviços IBM são projetados para fazerem parte de uma abordagem de segurança abrangente, que, necessariamente, envolverá procedimentos operacionais adicionais e poderá precisar de outros sistemas, produtos ou serviços para se tornar mais efetiva. A IBM NÃO GARANTE QUE QUAISQUER SISTEMAS, PRODUTOS OU SERVIÇOS SÃO IMUNES, OU DEIXARÃO SUA EMPRESA IMUNE DE CONDUTAS ILEGAIS OU MALICIOSAS DE QUALQUER PARTE.

Capítulo 1. Introdução

Este manual é uma introdução simples para um sistema simples. Não se engane; a palavra *simples* é usada aqui em seu contexto mais positivo e eficaz, pois a melhor maneira de você compreender um problema complexo é simplificá-lo; quebrá-lo em partes mais gerenciáveis e, então, dominar a fundo as partes constituintes. Dividir e conquistar. Esta é uma técnica que você usa instintivamente para resolver problemas diários e que é igualmente relevante para troca de informações de engenharia em um departamento, uma empresa ou no globo.

Se estiver impaciente para iniciar rapidamente a extração de informações de arquivos, de diretórios, de bancos de dados ou do Lotus Notes e a transferência desses dados para algum local diferente, você poderá desejar ir diretamente para o apêndice, “Guia do EasyETL”, na página 89. Este recurso permite tirar o maior proveito do IBM Security Directory Integrator sem precisar primeiro aprender conceitos importantes. Em vez disso, escolha sua origem e destino e, em seguida, pressione Executar e veja o seu fluxo de dados. Se, por outro lado, você desejar ter mais controle sobre como os dados são lidos, filtrados, aprimorados, transformados e movidos, continue fazendo a leitura aqui; o “Guia do EasyETL”, na página 89 ainda estará presente para o seu prazer de leitura futura.

IBM Security Directory Integrator¹ foi projetado e construído com a premissa de que mesmo os problemas de integração mais complexos podem ser decompostos em três partes básicas:

- Os sistemas envolvidos na comunicação – também chamados de *origens de dados*,
- Os *fluxos de dados* entre esses sistemas,
- Os *eventos* que acionam os fluxos de dados.

Com o IBM Security Directory Integrator, você converte o entendimento atômico do problema de integração diretamente em uma solução, construindo-a incrementalmente, um fluxo por vez, com feedback e verificação contínuos. Essa abordagem facilita a estimativa e o planejamento dos projetos de integração, reduzindo às vezes esse esforço à contagem e à determinação de custo de fluxos de dados individuais a serem implementados. A conclusão de uma tarefa em etapas executáveis também permite demonstrar regularmente o progresso às partes interessadas.

O IBM Security Directory Integrator acelera ainda mais o desenvolvimento, simplificando as diferentes técnicas entre suas origens de dados, permitindo que você gaste mais tempo concentrando-se nos requisitos dos negócios.

Potencializando o Eclipse, o ambiente de desenvolvimento do IBM Security Directory Integrator é abrangente e extensível. Os projetos de integração resultam em bibliotecas de componentes e em lógica de negócios que podem ser rapidamente reutilizadas para tratar novos desafios. Como resultado, as equipes em sua organização podem compartilhar ativos do IBM Security Directory Integrator, resultando em projetos independentes – até mesmo soluções de pontos específicos – que se ajustam imediatamente a uma infraestrutura integrada e gerenciada de forma coerente.

1. Não deixe se enganar pelo nome; O IBM Security Directory Integrator não está limitado a trabalho de diretórios e suporta todos os principais armazéns de dados, transportes, protocolos e APIs – incluindo, naturalmente, os diretórios LDAP.

Este documento apresenta uma introdução à metodologia de simplificação e resolução descrita acima. Você também executará suas primeiras etapas no sentido de explorar a apurada simplicidade do conjunto de ferramentas do IBM Security Directory Integrator, especificamente estes dois programas:

- O ambiente de desenvolvimento, chamado de *Configuration Editor*, ou 'CE' na forma abreviada,
- O ambiente de tempo de execução, simplesmente chamado de *Server*.

Você montará suas soluções do IBM Security Directory Integrator com o CE, enquanto um ou mais Servers serão usados para potencializá-los. Esses programas funcionam de forma coordenada, tornando a experiência do usuário perfeita, além de permitir que você trabalhe em várias plataformas; por exemplo, desenvolvendo em seu laptop enquanto testa e depura as soluções em execução remotamente em um mainframe.

Script em JavaScript

Conforme mencionado acima, o IBM Security Directory Integrator permite montar rapidamente as soluções de integração. Entretanto, para estender a funcionalidade automatizada e integrada com seu próprio processamento customizado e comportamento de fluxo, você precisará gravar trechos de script.

O script é feito em JavaScript e o IBM Security Directory Integrator inclui o IBM JSEngine para fornecer um ambiente de script rápido e confiável. Como resultado, você precisará usar e entender a linguagem JavaScript de núcleo. Há vários recursos on-line e em cópia impressa que são bons para aprender JavaScript. Verifique os grupos de notícias e os Web sites do IBM Security Directory Integrator para recomendações e links.

Para obter informações sobre script no IBM Security Directory Integrator, consulte o *Configurando o Directory Integrator*.

Instalando o IBM Security Directory Integrator

O IBM Security Directory Integrator é instalado em alguns minutos e você pode iniciar a construção, o teste e a implementação de soluções imediatamente. Ele é executado em uma grande variedade de plataformas, incluindo Microsoft Windows, IBM AIX, IBM System z e vários ambientes UNIX e Linux.

Há três caminhos de interesse ao instalar o IBM Security Directory Integrator e o instalador solicitará a você para especificar os dois primeiros:

1. O *Diretório de Instalação*, em que os arquivos de programa são mantidos, juntamente com os arquivos em lote ou scripts usados para ativar as diversas ferramentas.
2. O *Diretório de Solução*, geralmente abreviado como 'SolDir', que é a pasta atual sempre que você executa o IBM Security Directory Integrator. Você observará que os arquivos em lote e scripts de inicialização para o ambiente de desenvolvimento do Config Editor (`ibmditk`) e do Server (`ibmdisrv`) iniciam, ambos, um comando para alterar o diretório para o Diretório de Solução. Como resultado, todos os caminhos relativos em sua solução serão expandidos de seu Diretório de Solução.

3. A pasta *área de trabalho*. Este é o local onde seus arquivos de projeto e de recurso² são mantidos. Isto assumirá por padrão uma pasta nomeada "workspace" em seu Diretório de Solução.

Para obter informações adicionais sobre como instalar o IBM Security Directory Integrator, consulte *Instruções de instalação do IBM Security Directory Integrator no Instalando e administrando*.

Instalando os arquivos tutoriais

Os exercícios do tutorial neste manual requerem suporte a arquivos de dados que estão localizados na subpasta `examples/Tutorial` do diretório de instalação do IBM Security Directory Integrator. Por exemplo, uma instalação padrão do Windows colocaria esses arquivos no seguinte diretório:

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

O diretório 'Tutorial' deverá conter os seguintes arquivos:

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

Nota: Conforme mencionado na seção anterior, o instalador solicitará a você para especificar o local de seu Diretório de Solução. Esse será o local no qual os arquivos do projeto e de recursos serão armazenados e será geralmente um subdiretório chamado `My Documents\TDI` em sua área inicial.

Copie a pasta `Tutorial` para seu Diretório de Solução para torná-lo prontamente acessível a partir do conjunto de ferramentas do Configuration Editor.

Simplificar e solucionar

Esta seção ajuda você a entender sua posição inicial ao projetar uma solução de integração de dados. Embora a estratégia de design seja incremental, ela é apropriada para qualquer tamanho de projeto de integração e de implementação de sistemas, incluindo os grandes.

Use uma estratégia como esta para projetar uma solução de integração de dados, etapa por etapa, usando o IBM Security Directory Integrator:

- Reduza a complexidade fragmentando o problema em partes menores e gerenciáveis.
- Inicie com uma parte da solução geral, preferivelmente uma que possa ser concluída em uma semana ou duas.
- Inicie com uma parte da solução geral que possa ser colocada em produção totalmente sozinha.

2. Estes termos são explicados no Capítulo 2, "Introdução ao IBM Security Directory Integrator", na página 11

Manipulando Projetos Complexos

A melhor abordagem para compilar grandes projetos de integração e de implementação de sistemas é reduzir a complexidade dividindo o problema em partes menores mais gerenciáveis. Uma vez feito isso, você começa a trabalhar com uma parte da solução geral, preferivelmente uma que possa ser implementada independentemente. Desta maneira, ela já está fornecendo retorno sobre investimento enquanto você tenta resolver o restante.

Após isolar a parte com a qual você trabalhará, simplifique-a ainda mais mantendo o foco sobre as unidades básicas de comunicação: os próprios fluxos de dados. Agora, você está equilibrado para iniciar a implementação. O desenvolvimento de integração é feito usando o IBM Security Directory Integrator Configuration Editor (abreviado como 'CE') em uma série de ciclos tentar-testar-refinar, tornando o processo iterativo e até mesmo exploratório. Isso não somente ajuda você a descobrir mais sobre sua própria instalação, como também permite evoluir sua solução de integração à medida que seu entendimento do conjunto de problemas e seu impacto na infraestrutura crescem.

Tópicos Relacionados

Consulte os tópicos a seguir para obter uma explicação de como o IBM Security Directory Integrator permite transformar dados usando AssemblyLines.

- “Arquitetura do Kernel/Componente”
- “Modelo de Dados de Entrada-Atributo-Valor” na página 5
- “Fluxos de Dados = AssemblyLines” na página 6

Arquitetura do Kernel/Componente

Uma qualidade fundamental do IBM Security Directory Integrator é seu design de kernel/componente.

O termo *kernel* refere-se aqui à estrutura Rapid Integration Development (RID), que permite que você monte rapidamente suas soluções de integração e fornece uma lógica de execução automatizada para acioná-las. Os recursos que você precisaria codificar manualmente (e, portanto, são geralmente ignorados), como módulos de log/rastreamento, recuperação de conexão, detecção de mudança, manipulação de erros e uma API de gerenciamento externo, estão imediatamente disponíveis até mesmo para o fluxo de dados mais simples.

Além dessa funcionalidade de kernel genérica, o IBM Security Directory Integrator fornece um conjunto de componentes específicos da origem de dados: objetos auxiliares que simplificam os detalhes técnicos da integração com suas origens de dados. Os dois tipos de componentes que mais serão usados são *Conectores* e *Analísadores*.

Os conectores fornecem conectividade com uma grande variedade de origens de dados, além de herdarem a manipulação de dados estruturados, independentemente de sua organização subjacente. Alguns Conectores também servem como manipuladores de eventos, por exemplo, fazendo a ligação com portas IP e aguardando as conexões recebidas, ou 'atendendo' a mudanças que ocorrem em diretórios, bancos de dados ou arquivos.

Por outro lado, os Analísadores são usados para lidar com dados não estruturados – ou seja, fluxos de bytes, como aqueles localizados em arquivos, e-mail POP3/SMTP, mensagens MQ e fluxo de dados entre as várias portas IP.

O IBM Security Directory Integrator fornece uma biblioteca extensível de Conectores e Analisadores, cada um deles projetado para trabalhar com um sistema, serviço, API, transporte ou formato específico. A natureza intercambiável dos componentes do IBM Security Directory Integrator permite construir uma solução com base nos dados de teste – por exemplo, arquivos de texto – e, então, simplesmente descarregar para a área de troca os Conectores usados, a fim de apontar sua solução em origens ativas para verificação e implementação.

Além disso, os componentes do IBM Security Directory Integrator são fáceis de serem usados, além de serem fáceis de serem construídos e estendidos. Você pode aumentar sua biblioteca para lidar com origens de dados e serviços customizados, transferindo por download novos componentes de um Web site de comunidade, escrevendo seus próprios componentes em Java™ ou construindo e testando-os interativamente usando script diretamente no CE.

Modelo de Dados de Entrada-Atributo-Valor

A maneira como os dados são organizados e armazenados difere muito de um sistema para outro:

- Os bancos de dados armazenam informações em linhas, geralmente com um número fixo de colunas, cada uma transportando um único valor para esse registro;
- Os diretórios mantêm entradas orientadas a objetos que podem conter um número variado de atributos. Estes, por sua vez, contêm zero, um ou múltiplos valores³;
- Os bancos de dados do Lotus Domino contêm Documentos que são constituídos de Campos, sendo que cada um deles pode ser definido com valor único ou com diversos valores;
- Ainda, outros sistemas representam o conteúdo de dados como nós, objetos, registros, fluxos de bytes formatados ou conjuntos de valores das chaves.

Para que a comunicação seja significativa para todos os participantes, os formatos de dados precisam ser compatíveis ou devem ser convertidos para adequar cada sistema envolvido. Isso é chamado de *serialização de dados* e é geralmente a primeira dificuldade com que um especialista de integração se depara – e que pode consumir rapidamente uma parte dimensionável de recursos do projeto a serem superados. O IBM Security Directory Integrator trata disso automaticamente convertendo tipos específicos de origem em uma representação canônica consistente. Os valores dos dados são convertidos em objetos Java relevantes, com tipos nativos comparáveis sendo representados da mesma maneira. Por exemplo, as linhas lidas dos arquivos, os atributos de cadeia LDAP, os campos de texto do Domino e as colunas RDBMS CHAR e VARCHAR são todos convertidos em `java.lang.String` por seus respectivos Conectores.

Esses valores serializados são, então, acumulados em Atributos: objetos Java especializados definidos pelo IBM Security Directory Integrator. Conforme indicado acima, algumas origens permitem somente um único valor por coluna ou campo, enquanto outras permitem que vários valores sejam armazenados sob o mesmo nome de atributo. O Atributo do IBM Security Directory Integrator suporta implementações com valor único e com diversos valores e pode até mesmo não conter nenhum valor se necessário, por exemplo, ao representar uma coluna anulável em um banco de dados.

3. Considere por um momento o fato de que você provavelmente possui múltiplos endereços de e-mail, os quais podem ser todos armazenados no atributo de multivalores intitulado 'mail' no diretório de funcionários de sua empresa

Todos os Atributos que compõem uma única unidade de dados (ou seja, registro, mensagem, documento e outros) são coletados em outro objeto do IBM Security Directory Integrator chamado *Entrada*. Uma Entrada pode conter qualquer número de Atributos, ou nenhum.

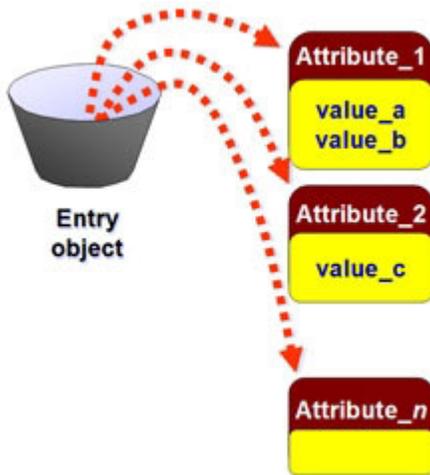


Figura 1. Um Modelo de Dados de Entrada-Atributo-Valor

Cada fluxo de dados possui um 'depósito' de Entradas principal chamado de Entrada de Trabalho. Sempre que um Conector lê os dados, ele cria Atributos e os coloca na Entrada de Trabalho. Qualquer Conector configurado para saída usa os Atributos já localizados na Entrada de Trabalho para acionar mudanças nos sistemas de destino.

Essa abordagem de dois estágios fornece flexibilidade quase ilimitada em como os dados são transferidos, transformados, filtrados e aprimorados. Isto também significa que você pode construir inicialmente seu fluxo de dados inteiramente com Conectores de entrada e, então, examinar os dados interativamente com o CE enquanto eles são lidos e manipulados, antes até mesmo que você tenha que considerar conexões com sistemas de saída.

Conforme você verá posteriormente, a Entrada do IBM Security Directory Integrator manipula dados hierárquicos complexos tão facilmente quanto ele faz para um esquema simples.

Fluxos de Dados = AssemblyLines

Cada fluxo de dados em sua solução é implementado como um *AssemblyLine* do IBM Security Directory Integrator, também abreviado como 'AL' nesta e em outras publicações.

ALs são listas ordenadas de componentes que formam um caminho único e contínuo de origens de entrada para destinos. O comportamento integrado fornecido pelo kernel junta os componentes e passa os dados transportados na Entrada de Trabalho de um para outro.

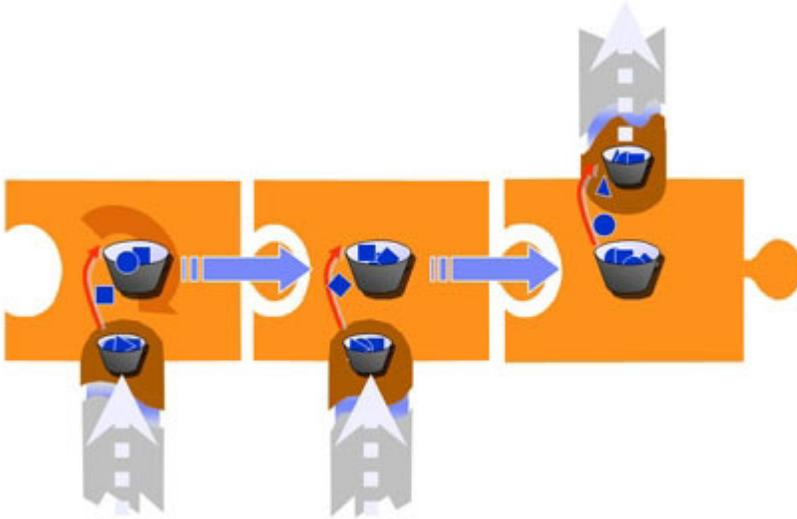


Figura 2. Dados Fluindo em um AssemblyLine

Dizem que uma figura vale mais que mil palavras e o diagrama acima não é uma exceção. As três peças do quebra-cabeça representam Conectores vinculados para formar um AssemblyLine. A 'base' mais escura de cada peça do quebra-cabeça realça a parte específica da origem de dados do Conector – ou seja, a *interface* para o sistema conectado – conhecida como *Connector Interface* (abreviada como 'CI'). O restante de cor mais clara de cada peça do quebra-cabeça representa a funcionalidade genérica do kernel que faz todos os componentes funcionarem de um modo similar e previsível, possibilitando que eles sejam vinculados e fornecendo padrões automatizados de comportamentos com pontos de controle para customização⁴.

Esta figura ilustra alguns conceitos mais importantes. Por exemplo, além da Entrada de Trabalho mostrada acima fluindo de um componente para outro no AssemblyLine, há um "depósito" Java adicional em cada Connector Interface. Cada objeto de Entrada local é usado para armazenar os dados em cache durante as operações de leitura e gravação executadas por essa CI e é chamado de *Entrada Conn.*

Agora observe as setas curvadas que ilustram o fluxo de dados entre as várias Entradas Conn e a Entrada de Trabalho do AL. Estes são *Mapas de Atributos* e cada um representa um conjunto de regras para movimentação e transformação de dados em seu caminho dentro ou fora do AL. Aqueles que elevam dados de uma Entrada Conn para uma Entrada de Trabalho são nomeados *Mapas de Entrada*, pois determinam quais dados são trazidos para o AssemblyLine. A seta na peça do quebra-cabeça mais à direita que mostra os dados se movimentando na outra direção – da Entrada Work para a Entrada Conn – é chamada de *Mapa de Saída*.

Como há somente uma Entrada de Trabalho em qualquer tempo, você pode deduzir que os AssemblyLines processam um item por vez: por exemplo, uma linha do banco de dados, uma entrada de diretório, uma mensagem do MQ e assim por diante. Este é um outro aspecto importante do IBM Security Directory

4. Como você pode ver, todo componente do AssemblyLine reflete a arquitetura do kernel/componente do IBM Security Directory Integrator. Se você decidir fazer seu próprio componente, somente sua interface precisará ser implementada. O "wrapper" do AL e sua rica funcionalidade integrada estão disponíveis automaticamente, como cortesia do kernel do IBM Security Directory Integrator.

Integrator e, embora um AssemblyLine possa efetuar o ciclo de centenas ou até mesmo milhares de Entradas por segundo⁵, é uma consideração importante ao projetar sua solução. Naturalmente, é possível propagar o trabalho para múltiplos AssemblyLines. Você localizará esta e outras técnicas para otimizar o desempenho do AL em outras publicações do IBM Security Directory Integrator.

Introdução

Um bom início para qualquer projeto de integração é fazer um diagrama do problema à mão.

Usando um lápis e um pedaço de papel, faça um esboço dos fluxos desejados em traços largos. Este exercício não somente ajuda a visualizar o escopo da tarefa, como também serve como um projeto para implementar esses fluxos no IBM Security Directory Integrator.

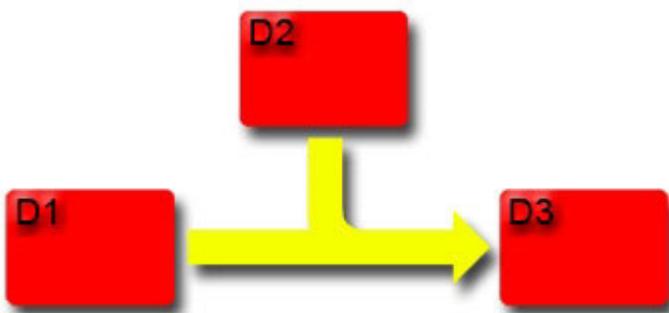


Figura 3. Cenário do Tutorial

A primeira etapa na criação de uma solução do IBM Security Directory Integrator é converter fluxos de dados entre origens de dados em AssemblyLines constituídos de Conectores. O mantra 'simplificar e resolver' do IBM Security Directory Integrator prescreve a construção de sua solução incrementalmente, iniciando da forma mais simples possível.

Para ilustrar isso, considere o cenário de exemplo que será usado para seu primeiro AssemblyLine. Esta tarefa de integração envolve três origens de dados, rotuladas como D1, D2 e D3. A solução desejada é migrar o conteúdo de D1 para D3, aumentando esses dados com valores localizados em D2. Convertendo esse requisito para um AssemblyLine, você acaba com três Conectores, um para cada origem de dados:

1. o primeiro Conector para *iterar* através de D1, alimentando esses dados no fluxo;
2. seguido por um segundo Conector que *consulta* os registros relacionados em D2 e mescla estes valores com aqueles provenientes de D1;
3. por último, um terceiro Conector configurado para *incluir* esses registros aumentados para D3.

Em vez de combater o problema inteiro de uma vez, o IBM Security Directory Integrator permite simplificar a tarefa, iniciando com somente dois Conectores: um que lê o conteúdo de D1 no AL e outro para gravar esses valores em D3. Assim que o AssemblyLine mínimo estiver funcionando corretamente, ele poderá ser

5. O desempenho dependerá do design e da complexidade do AssemblyLine e da configuração da máquina que está executando o Servidor.

estendido com o Conector no D2 para juntar-se a Atributos adicionais. É exatamente desta forma como você criará sua primeira solução do IBM Security Directory Integrator e as etapas para guiá-lo por esse processo constituem o restante deste guia.

Capítulo 2. Introdução ao IBM Security Directory Integrator

Esta seção fornece informações úteis para a compreensão dos princípios básicos do IBM Security Directory Integrator, além de um conjunto de exercícios do tutorial para proporcionar experiência prática com o ambiente de desenvolvimento.

Sua primeira etapa para conhecer o produto é iniciar o conjunto de ferramentas de desenvolvimento do IBM Security Directory Integrator, conhecido como Configuration Editor, ou CE na forma abreviada.

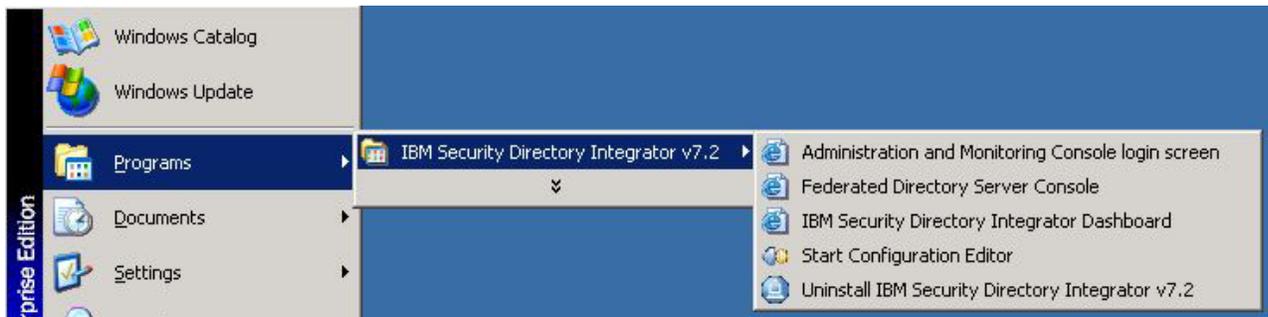


Figura 4. Iniciando o Configuration Editor

Na primeira vez em que inicializar o CE, você verá este diálogo para especificar sua área de trabalho.

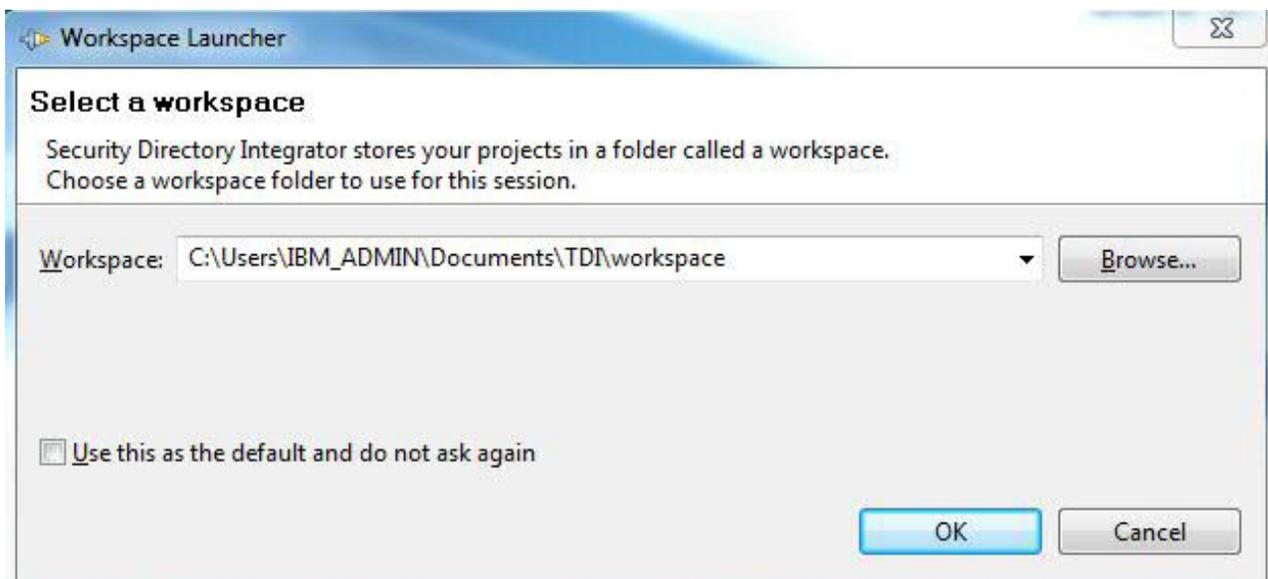


Figura 5. Selecionando sua Área de Trabalho

Sua área de trabalho é onde o Configuration Editor armazenará seus arquivos de projeto, incluindo componentes e AssemblyLines, e está geralmente localizada sob seu Diretório de Solução.

Uma vez que você esteja satisfeito com o local de sua área de trabalho, pressione o botão **OK**. Agora, a Tela de Boas-vindas aparecerá.

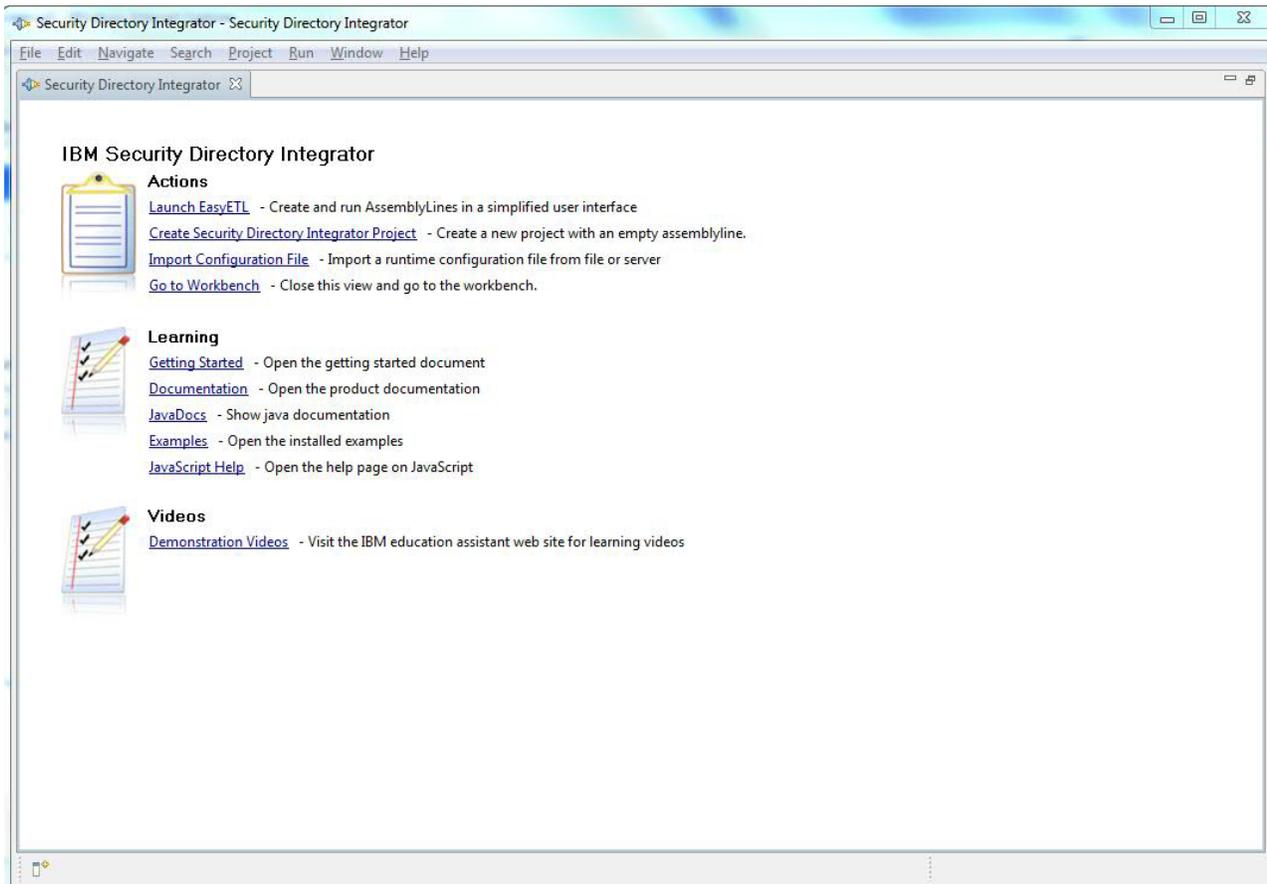


Figura 6. Tela de Boas-vindas do Configuration Editor

A tela de Boas-vindas oferece vários links de iniciação rápida⁶.

Sempre que constrói, testa ou modifica as soluções de integração com o IBM Security Directory Integrator, você está trabalhando em um projeto. Os projetos são coletas de AssemblyLines e seus componentes constituintes, e cada projeto aparece em sua própria subpasta de sua área de trabalho. Os AssemblyLines e os componentes que constituem um projeto são armazenados como arquivos individuais, que por sua vez estão localizados em subdiretórios na pasta do projeto.

Selecione o segundo link na parte superior desta página⁷ (*Criar Projeto do Security Directory Integrator*) para configurar seu primeiro projeto. Agora você deve fornecer um nome para seu novo projeto. Denomine-o como 'Tutorial' e pressione **Concluir**.

6. Você pode retornar a essa tela a qualquer momento selecionando **Ajuda > Bem-vindo** no Menu Principal.

7. O primeiro link, Ativar EasyETL, abre um ambiente de trabalho simplificado e é abordado no apêndice, “Guia do EasyETL”, na página 89.

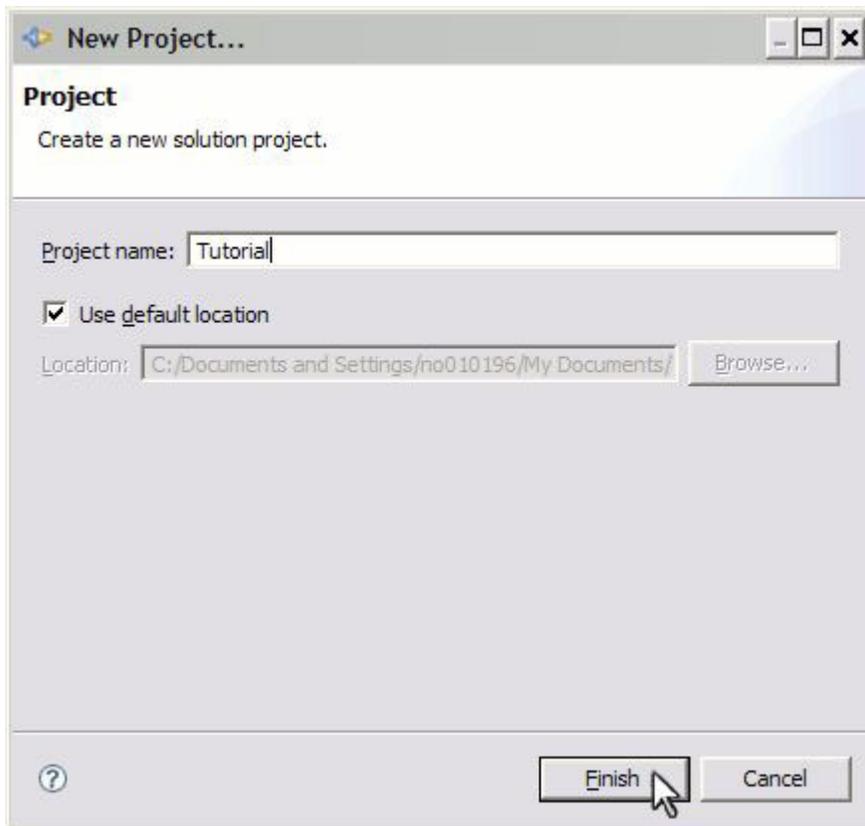


Figura 7. Nomeando seu Novo Projeto

Agora, você verá a área de trabalho de desenvolvimento principal. Aqui, os painéis podem ser todos redimensionados e você pode decidir como a tela é organizada. O que você vê aqui na tela é a Perspectiva do IBM Security Directory Integrator padrão⁸.

8. Uma *Perspectiva* é simplesmente uma organização dos painéis do ambiente de desenvolvimento. Se você fez mudanças no layout e desejar retornar à Perspectiva do IBM Security Directory Integrator padrão, simplesmente clique em **Janela** no menu superior e selecione a opção **Reconfigurar Perspectiva**.

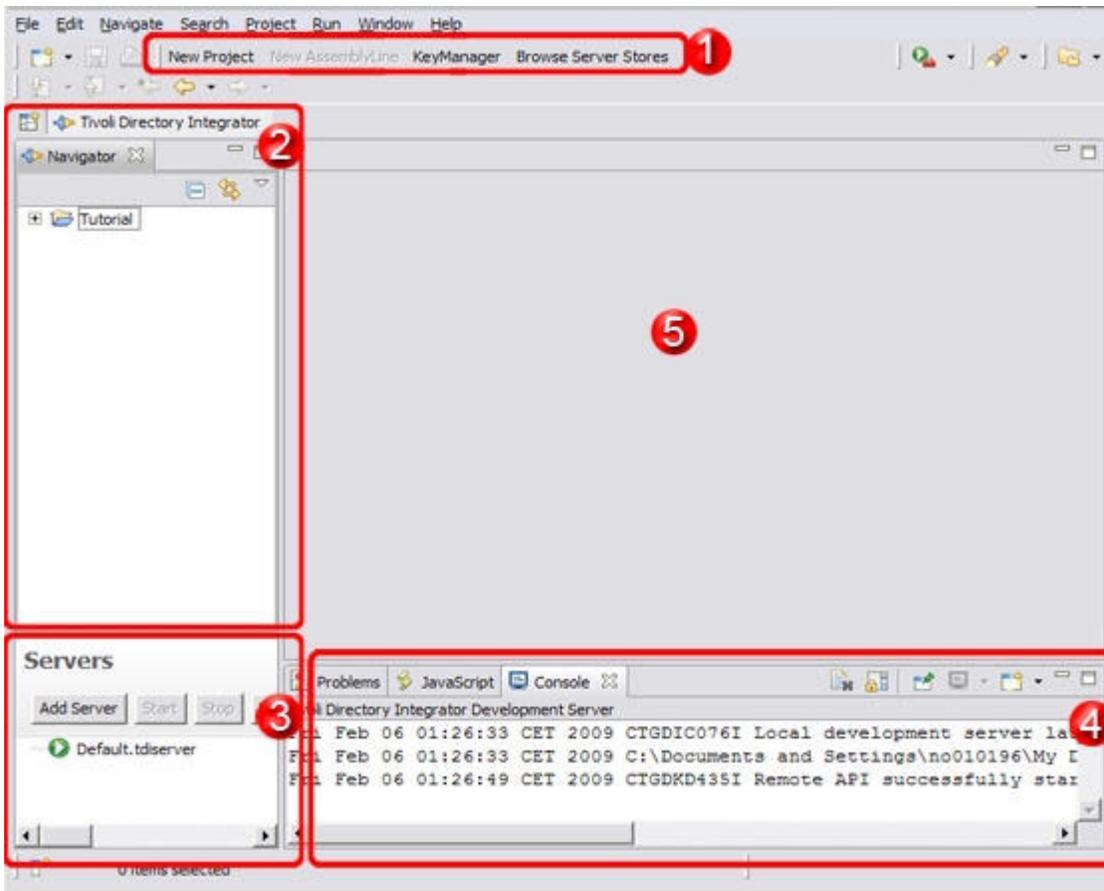


Figura 8. Tela Principal do Config Editor

Nota: A única perspectiva que você precisa usar é a perspectiva **Security Directory Integrator** ou **Easy ETL**. Se seu Editor de Configuração não tiver a aparência esperada ou não tiver áreas de janela esperadas, tente as etapas a seguir:

- Selecione **Janela > Abrir Perspectiva > Security Directory Integrator**. Essa opção seleciona a perspectiva Security Directory Integrator.
- Se isso não ajudar, selecione **Janela > Reconfigurar Perspectiva....** Esse comando assegura que todas as áreas de janela padrão estejam posicionadas como deveriam estar.

Esta é a tela principal na qual você gastará a maior parte de seu tempo quando trabalhar com o IBM Security Directory Integrator. Sem entrar nos detalhes de todos os elementos navegacionais aqui⁹, vamos examinar as área numeradas, realçadas na captura de tela acima:

1. No meio da linha do botão principal, há um conjunto de atalhos para criar novos Projetos e, se um Projeto for selecionado no Navegador, para criar novos AssemblyLines nele. Também há um botão para ativar a ferramenta KeyManager para trabalhar com keystores e armazenamentos confiáveis de certificados; assim como um botão **Procurar Armazenamentos do Servidor** para recuperar as várias configurações de propriedades do Servidor IBM Security Directory Integrator associado a esse Projeto.

9. Tal como ocorre com a maioria dos aplicativos baseados no Eclipse, haverá diversas maneiras de executar a mesma operação. O *Configurando o Directory Integrator* descreve todas as várias opções e painéis disponíveis.

2. Este é o painel do *Navegador* e fornece uma visualização em árvore de seus ativos de desenvolvimento. Seu novo Projeto 'Tutorial' deverá aparecer aqui.
3. O painel Servidores exibe o status de todos os Servidores configurados. É possível ver pelo ícone de seta próximo a 'Default.tdiserver' que esse Servidor foi iniciado para você. Esse painel também fornece botões para definir novos Servidores, iniciar e parar seus Servidores, além de atualizar a lista e visualizar o log de um Servidor¹⁰.

Observe que sempre que você ativa um AssemblyLine, a Instância de Configuração¹¹ e o AL também são mostrados nesse painel.

4. Aqui, você verá um conjunto de guias com a guia atualmente selecionada mostrando a saída do console proveniente de seu Servidor. As mensagens exibidas aqui indicam agora que seu Servidor está em execução e que sua API está inicializada e pronta para uso.
5. A área cinza nesta captura de tela é onde os painéis do *editor* aparecem, à medida que você cria e abre os AssemblyLines e os componentes. Cada tipo de recurso (Conector, Analisador, AssemblyLine e outros) possui seu próprio editor especialmente projetado.

Criando seu Primeiro AssemblyLine

Retornando ao cenário de exemplo descrito na introdução, agora você criará um AL que migra informações de D1 para D3, ignorando o momento da junção dos dados do D2.



Figura 9. Diagrama de Cenário Simplificado com Somente Duas Origens de Dados

A pasta 'Tutoriais' (que você deve ter copiado de *diretório de instalação do TDI/examples* para seu Diretório de Solução) contém um arquivo nomeado `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. Se por algum motivo seu servidor não tiver sido iniciado corretamente, abra 'Servidores TDI' e clique duas vezes em 'Default.tdiserver'. Isso abre o Documento do Servidor associado. Certifique-se de que as configurações do Diretório de Instalação e de Solução estejam corretas e, em seguida, pressione a opção **Criar Diretório de Solução** na parte superior desse painel. Se isso não corrigir o problema, entre em contato com o suporte.
 11. Sempre que o Servidor IBM Security Directory Integrator carrega uma Configuração, ele cria uma *Instância de Configuração* que encapsula os AssemblyLines desse projeto e permite que eles sejam executados em seus próprios ambientes contidos. Isto significa que você pode carregar a mesma Configuração múltiplas vezes no mesmo Servidor, resultando em Instâncias de Configuração separadas, todas contendo o mesmo conjunto de ALs, sem que estes interfiram uns com os outros.

Você pode observar na listagem acima que isto está no formato *Character Separated Value* (CSV). Esse arquivo representa nossa origem de dados de entrada D1. Seu AL extrairá esses dados e os transferirá para um documento XML, que será nosso destino de saída D3.

Clique em **Novo AssemblyLine** na barra de ferramentas superior e denomine o novo AL como 'CSV2XML'.

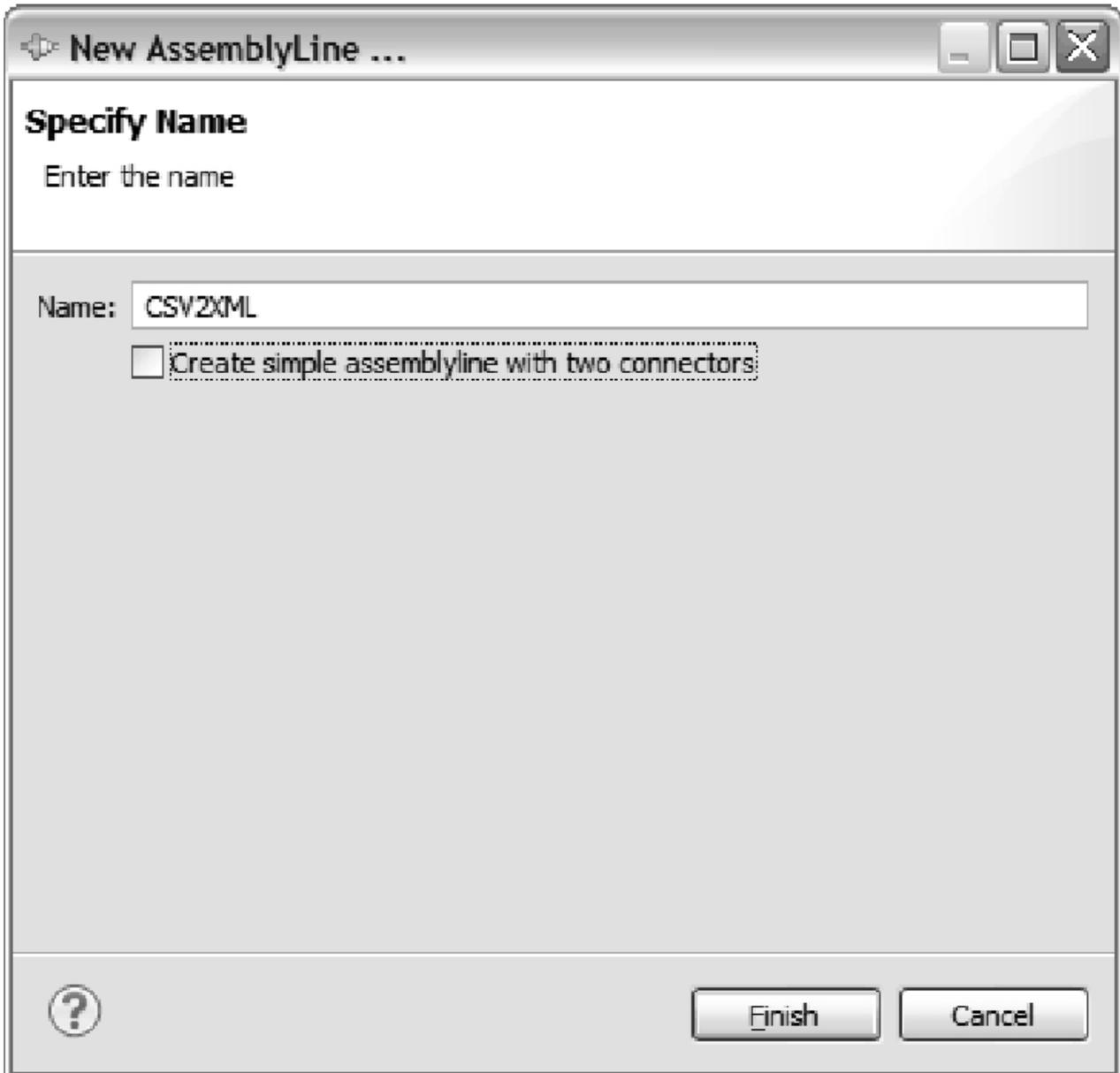


Figura 10. Caixa de Diálogo Novo AssemblyLine

Agora pressione o botão **Concluir** para abrir o AL em uma guia do editor de AssemblyLine.

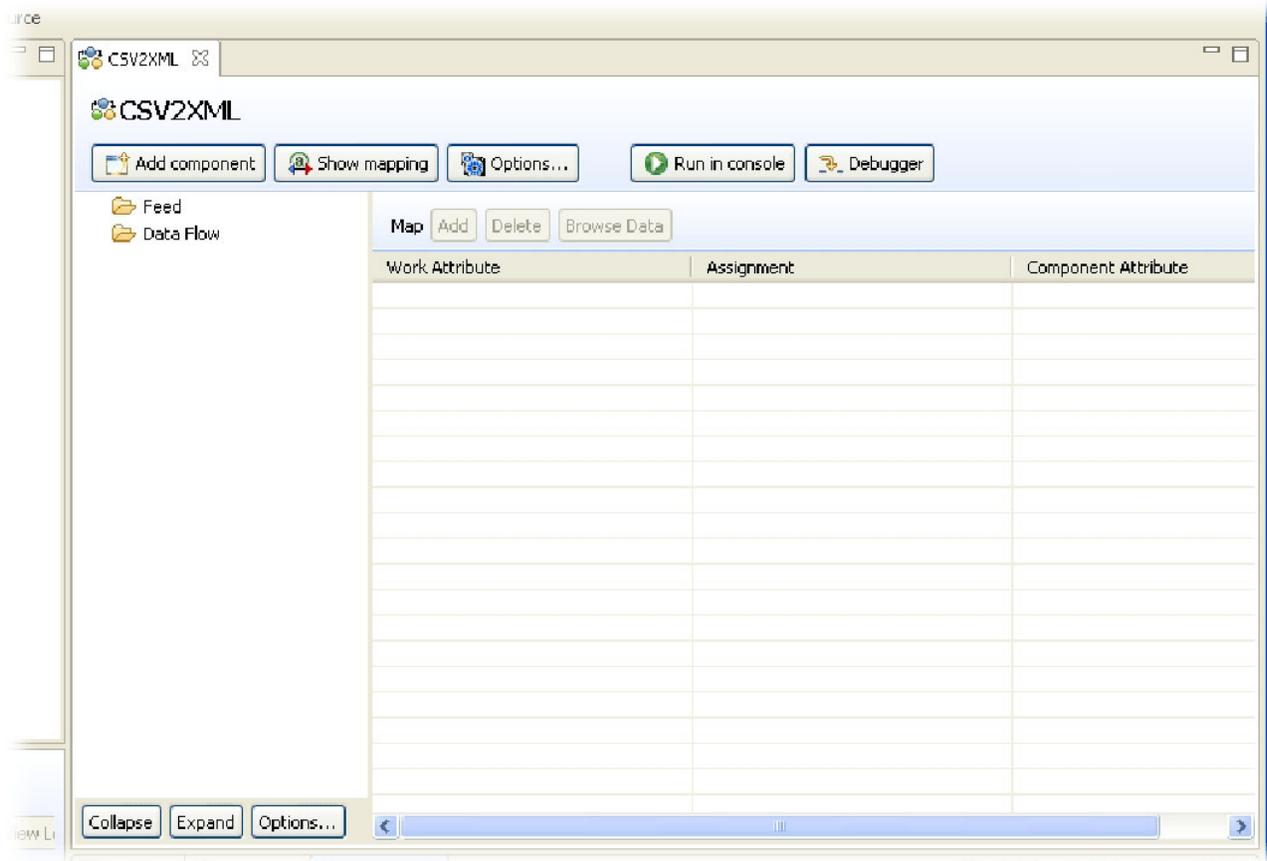


Figura 11. Editor de AssemblyLine Vazio

A parte esquerda do editor de AL contém a lista de componentes que compõem esse AssemblyLine e está vazia no momento, exceto para os nomes de seção: *Alimentação* e *Fluxo de Dados*. A área à direita exibe todos os Atributos que sendo mapeados dentro e fora do AL.

Para entender essas seções do AssemblyLine, considere por um momento o que queremos que esse novo AL execute: *Para cada linha no arquivo CSV, crie um novo nó no documento XML*. Esse comportamento em loop é fornecido automaticamente pelo kernel do IBM Security Directory Integrator, acionando os componentes listados na seção *Fluxo de Dados* do AL, contanto que existam dados de entrada provenientes de Conectores na seção *Feed*¹².

Vamos aproveitar a vantagem dessa funcionalidade incluindo um Conector na seção *Alimentação* para leitura em nosso arquivo de entrada CSV. Faça isso clicando com o botão direito do mouse na pasta da seção *Alimentação* e selecionando **Incluir Componente...**

12. Observe que somente um Conector *Feeds* entregará dados para o AL por vez. Se você colocar mais de um Conector de Agente Iterativo aqui, o que estiver em primeiro lugar será esvaziado primeiro, antes do próximo na linha iniciar a leitura a partir de sua origem.

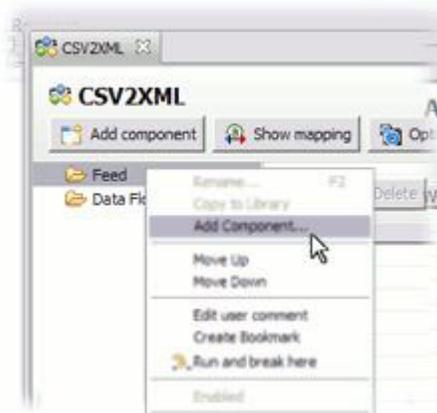


Figura 12. Inserindo um Novo Componente

O assistente **Escolher Componente** será apresentado.

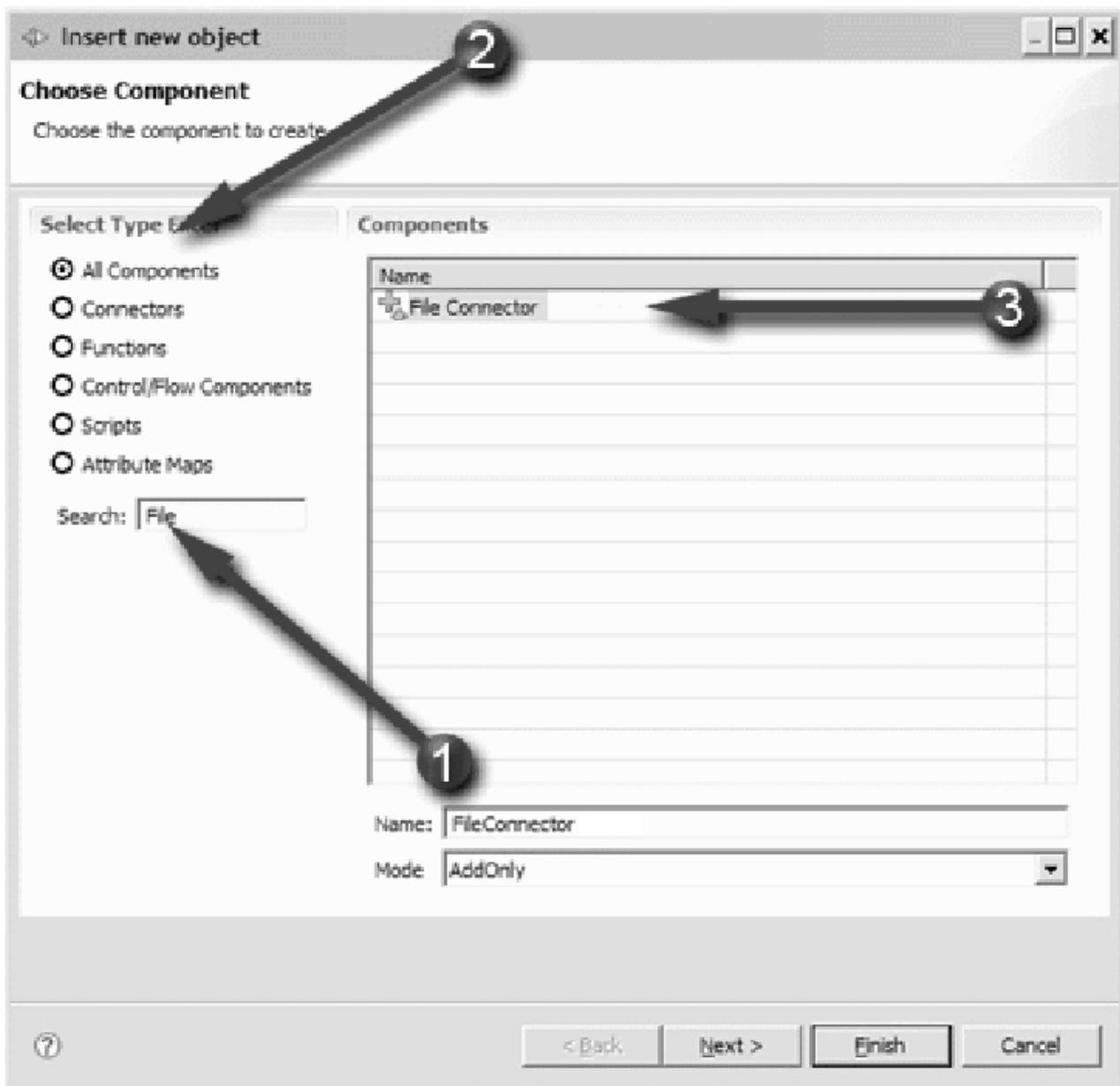


Figura 13. Escolhendo o Componente

Esse diálogo fornece várias opções para localizar e selecionar o componente desejado:

1. Inicie digitando qualquer parte do nome do componente no campo de texto e a lista de seleção à direita será filtrada apropriadamente. Para este exemplo, digite "file".
2. Opcionalmente, você pode limitar a lista de seleção para incluir somente um único tipo de componente – Conectores, Analisadores, Scripts e outros.
3. Localize e selecione o componente desejado nesta lista. No nosso exemplo, isso será 'Conector de Arquivo'.

O novo Conector será nomeado automaticamente como 'FileSystemConnector'. Altere-o para 'Read_CSV_File' para que tenha mais significado no contexto de sua

solução¹³ e, em seguida, selecione **Agente Iterativo** no menu suspenso **Modo**.

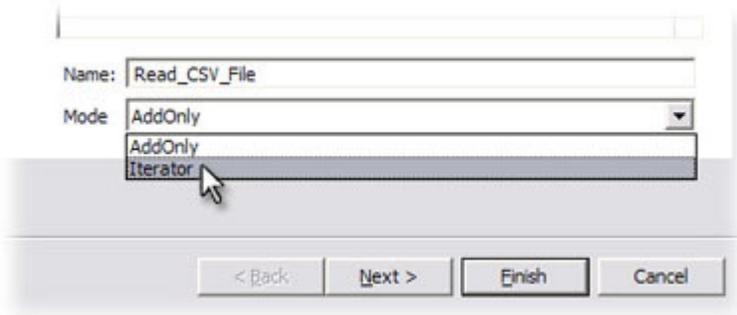


Figura 14. Renomeando o Conector e Alterando seu Modo

A configuração Modo de um Conector que indica à lógica de execução do AL integrado qual função esse componente desempenha no fluxo. O Modo de Agente Iterativo resulta no comportamento *for-each* que você precisa para acionar os dados a partir do arquivo CSV, uma entrada por vez, para os componentes que serão incluídos na seção *Fluxo de Dados*.

Agora, pressione o botão **Próximo** para continuar no painel de configuração para o Conector selecionado.

13. Embora você possa nomear Conectores conforme desejar, recomenda-se nomeá-los da mesma maneira que faria para uma variável do script: inicie com uma letra, seguida por qualquer número de letras, dígitos e caracteres de sublinhado. Isto porque todos os componentes do AL são registrados automaticamente como variáveis do script, tornando mais fácil se você posteriormente desejar reconfigurar e acioná-los a partir de seu código de script.

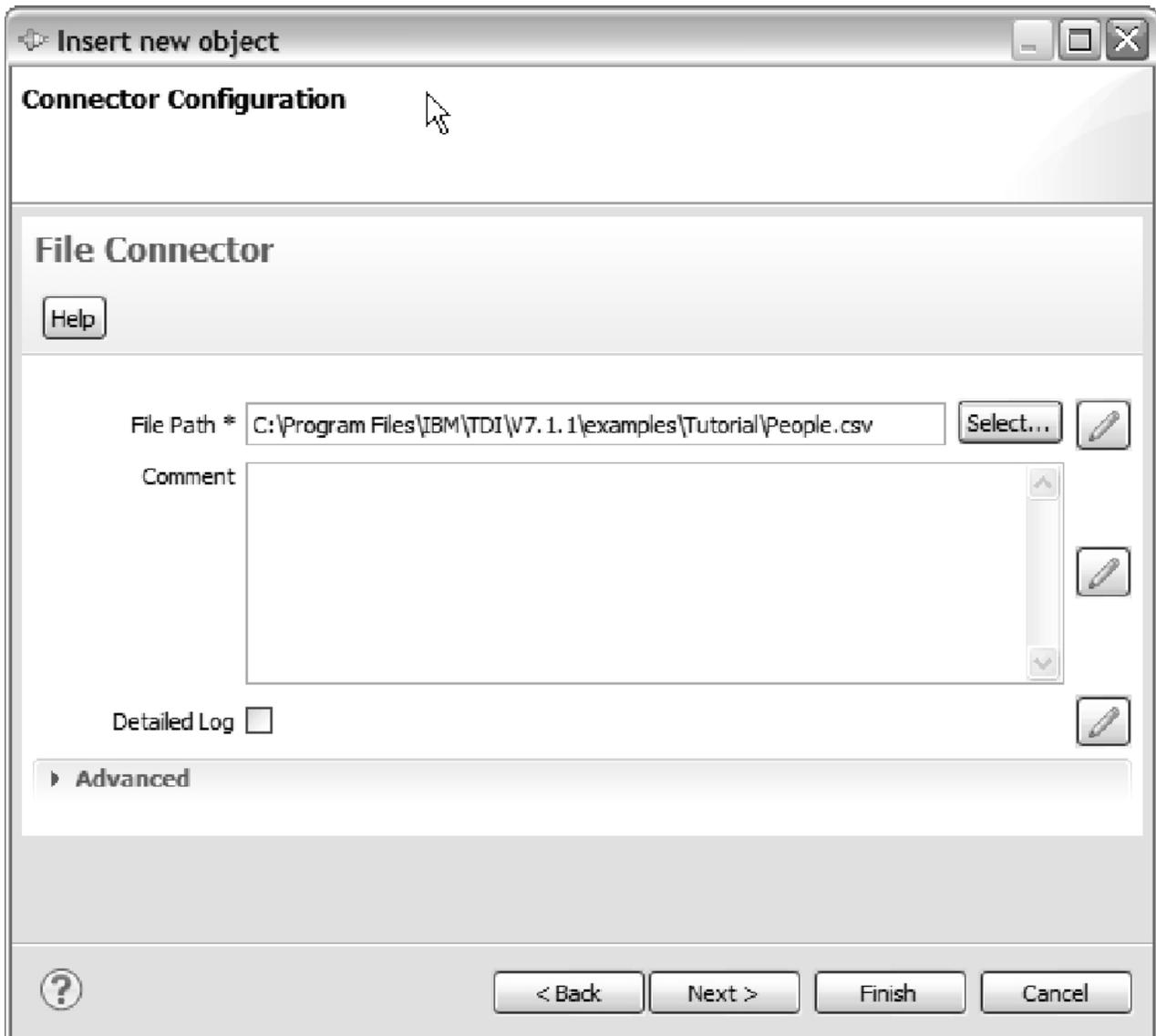


Figura 15. Painel de Configuração do Conector de Arquivo

Cada componente fornece seu próprio conjunto de parâmetros de configuração. Aqueles que são mostrados na tela são para o Conector de Arquivo e possui apenas um parâmetro necessário: **Caminho de Arquivo**. Digite no caminho no arquivo `People.csv`, seja o caminho completo ou o caminho relativo a partir do seu Diretório de Solução, como mostra a captura de tela acima¹⁴ – ou pressione o botão **Selecionar** para abrir um navegador de arquivos para localizar esse arquivo.

Como um arquivo de texto formatado é um fluxo de bytes e não uma origem de dados estruturados, como um banco de dados ou diretório, você deve configurar um Analisador para interpretar a formatação do fluxo enquanto ele é lido. O IBM Security Directory Integrator fornece um recurso de Navegador de Dados poderoso e versátil para testar interativamente sua seleção e configuração de Conector/Analisador.

14. Essa técnica facilita mover e compartilhar sua solução, porque basta especificar o Diretório da Solução que desejar para que todos os caminhos relativos funcionem inalterados.

Examinaremos isso daqui a pouco, mas primeiro você precisa completar esse assistente pressionando **Próximo** novamente e continuando com **Configuração do Analisador**. Agora, você clicará no **Analisador CSV** para selecioná-lo.

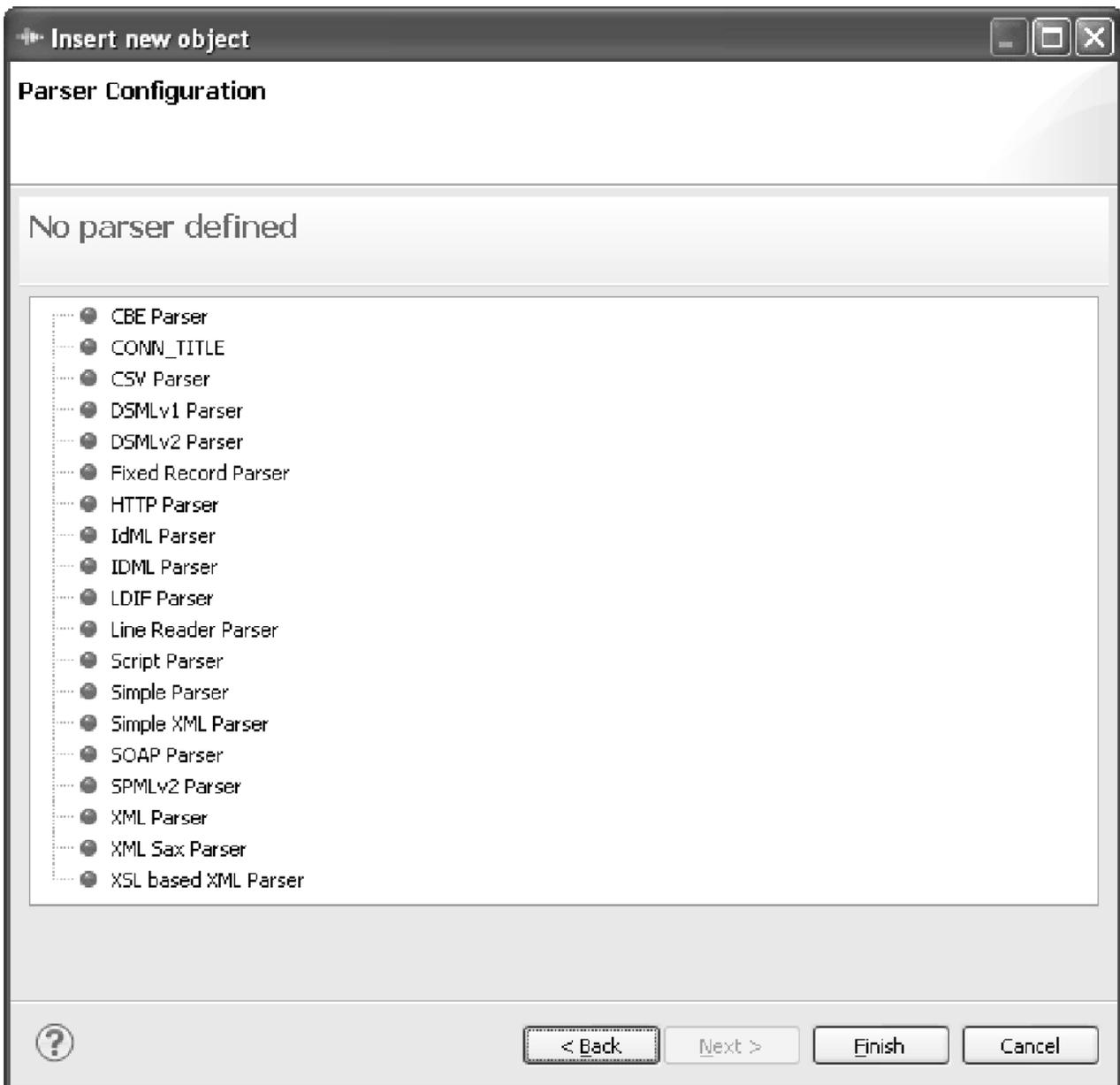


Figura 16. Selecionando o Analisador Durante Inserir Novo Objeto

Depois que você tiver selecionado o Analisador CSV, pressione **Concluir** para fechar o assistente. Agora, você verá o painel Configuração do Analisador. Como não é necessário alterar as configurações padrão, simplesmente pressione Concluir novamente para completar o assistente.

A próxima etapa é permitir que o Conector *descubra* o esquema de sua origem de entrada para mapear esses volumes para o AssemblyLine. É neste local que o

Navegador de Dados torna-se acessível¹⁵. Inicie com o Navegador de Dados clicando com o botão direito do mouse em seu novo Conector de Agente Iterativo na árvore de Componentes do AssemblyLine e selecionando **Pesquisar Dados** no menu de contexto.

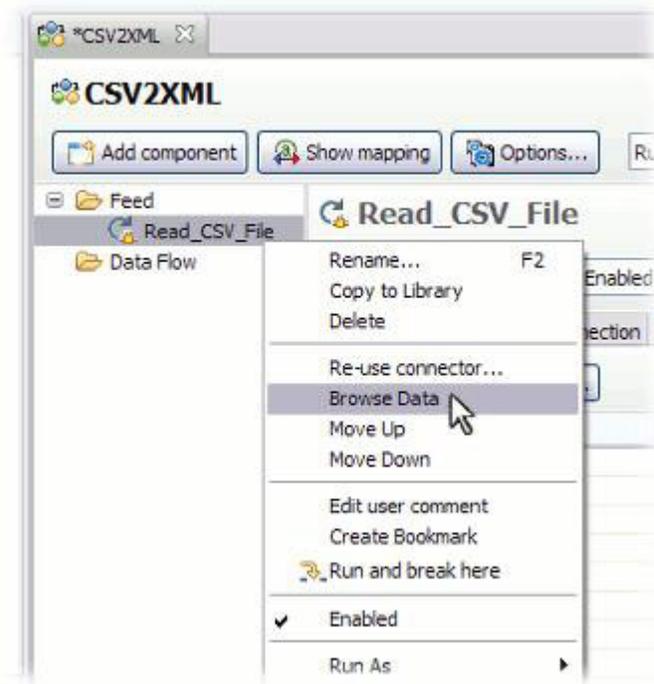


Figura 17. Seleção do Menu de Contexto Pesquisar Dados

Isso abrirá o Navegador de Dados em uma nova guia do editor.

15. Como você sabe que o arquivo está no formato CSV, a abordagem mais rápida seria apenas clicar no botão **Conectar** e **Próximo** na área Esquema do Conector de Agente Iterativo. Em seguida, arraste os Atributos descobertos para o Mapa de Entrada como desejar. O Navegador de Dados é útil quando você não tem certeza sobre o formato. Mesmo assim você deveria tentá-lo :)

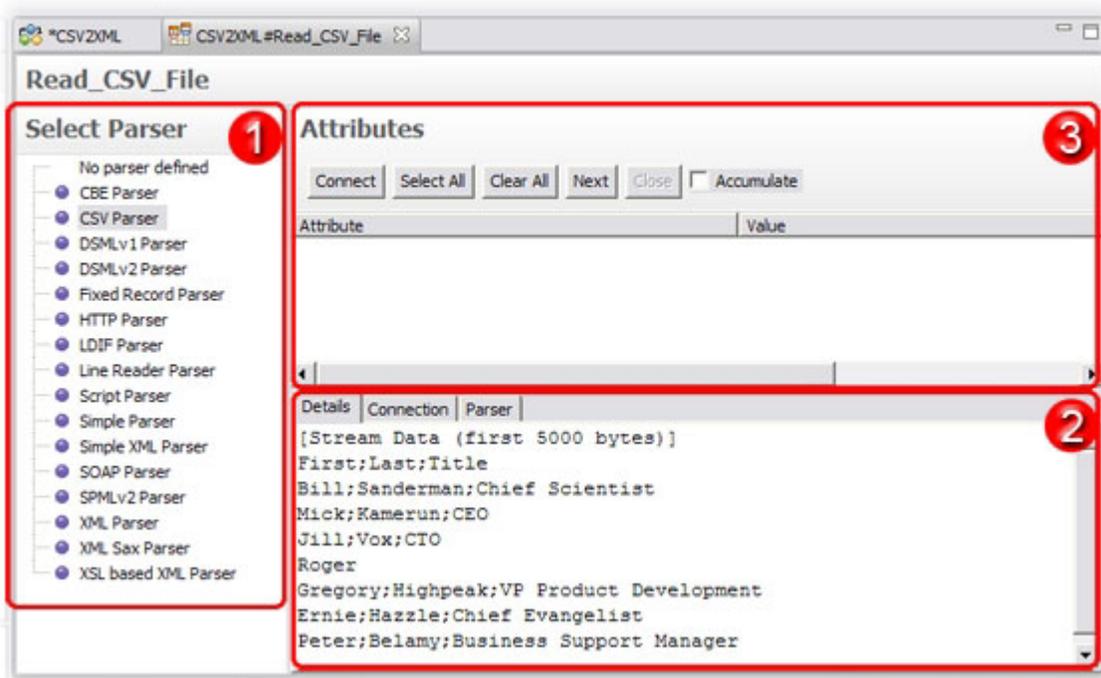


Figura 18. O Navegador de Dados

A área rotulada 1 na captura de tela acima é para escolher – e alterar – o Analisador selecionado. A área 2 fornece uma guia Detalhes que mostra o fluxo de bytes bruto que será analisado. Também há guias para alterar os parâmetros de Conexão do Conector, assim como uma para configurar o Analisador escolhido.

A última seção deste diálogo (#3 na captura de tela) é para conectar-se à origem de dados e descobrir quais Atributos estão disponíveis. Execute isso agora primeiro pressionando **Conectar** e, depois, o botão **Próximo**.

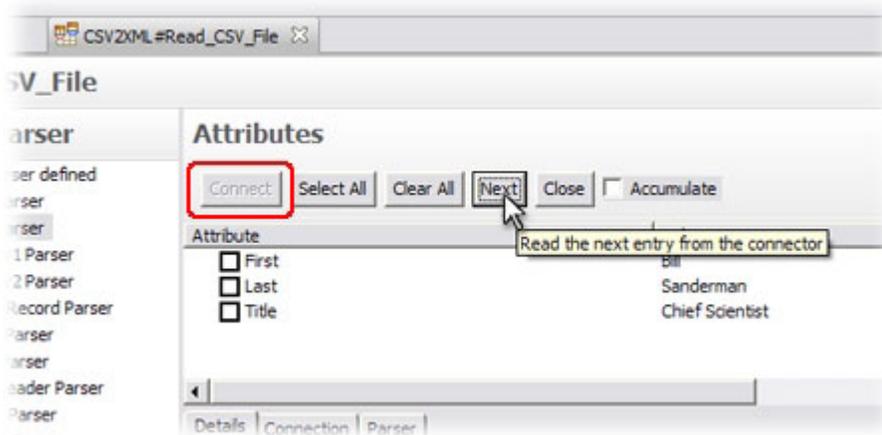


Figura 19. Descobrimo o Esquema Interativamente pela Pesquisa de Dados Ativos

Você acabou de descobrir o *esquema* desse arquivo. Selecione os Atributos que você deseja mapear, que neste caso são todos eles, selecionando a caixa de opção próxima a cada um deles ou usando o botão **Selecionar Todos**.

Use o atalho Ctrl-W para fechar a guia Navegador de Dados ou simplesmente clique no símbolo 'Fechar' (X) na extremidade direita da guia e retorne ao editor do AssemblyLine, onde o AL deverá ser semelhante à captura de tela abaixo¹⁶.

Figura 20. AL com Conector de Agente Iterativo no Local

Detalhes para o componente selecionado são mostrados à direita da lista de componentes do AL, incluindo as três regras de mapeamento que você acabou de configurar no Mapa de Entrada. Cada item do Mapa de Atributos possui uma **Designação**, que é um trecho de script avaliado a fim de configurar o valor (ou valores) do Atributo de destino.

Antes de continuar, dedique um momento para refletir sobre estas *Designações*: Você recordará a partir da seção “Modelo de Dados de Entrada-Atributo-Valor” na página 5 que o AssemblyLine possui uma *Entrada de Trabalho* globalmente disponível que transporta todos os dados que estão sendo transportados abaixo para o AL. Esse objeto é referenciado no código de script usando a variável de script pré-registrada *work*. Além disso, a Interface de cada Conector possui sua própria *Entrada Conn*, que é usada como um cache para leituras e gravações. Esse objeto específico do componente é acessado a partir do script por meio da variável pré-registrada *conn*¹⁷. Para ilustrar, considere a primeira regra de mapeamento. Cria-se um Atributo na Entrada de Trabalho nomeada 'First'. Seu valor é derivado da seguinte designação:

```
conn.First
```

Essa notação abreviada referencia o Atributo denominado 'First' que acabou de ser lido na Entrada *conn* e seus valores são usados para preencher o novo Atributo de Entrada de Trabalho. Um script de designação comparável seria:

16. Se por algum motivo seu Conector estiver na seção *Fluxo de Dados*, simplesmente arraste-o para cima até *Alimentação*. Se a configuração de modo não for Agente Iterativo, clique com o botão direito do mouse no Conector, selecione **Modo** e em seguida, escolha **Agente Iterativo**.

17. A variável *conn* está disponível somente por períodos limitados, conforme mostrado nos Fluxogramas de Ganchos do IBM Security Directory Integrator. Contudo, fora desse escopo, é possível acessá-la consultando um componente por sua Entrada *conn*.

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

Voltando ao exercício, agora você precisa incluir seu Conector de saída para criar o documento XML de destino (origem de dados D3). Desta vez, tente usar o botão **Incluir Componente** na parte superior do painel Componentes do AssemblyLine.

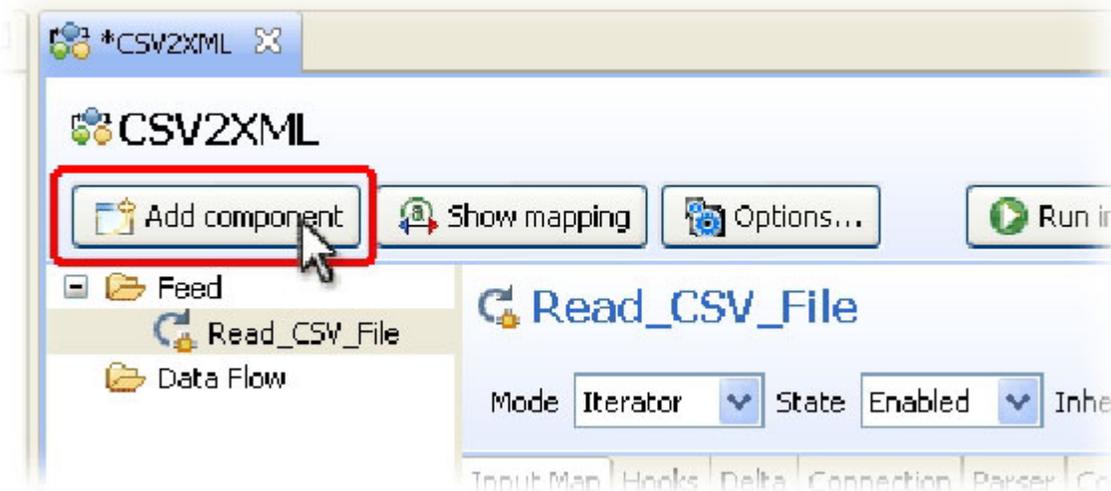


Figura 21. Botão Incluir Componente

Novamente, escolha o Conector de Arquivo renomeando-o para 'Write_XML_File'. Deixe a configuração de Modo como **AddOnly** e, em seguida, pressione **Próximo**.

No painel Configuração do Conector, configure o parâmetro **Caminho de Arquivo** para gravar em um arquivo chamado `Output.xml` na pasta Tutorial. Em seguida, escolha 'Analisador XML' no próximo painel do Assistente. Agora você pode pressionar **Concluir**, pois não precisa alterar a configuração do Analisador XML. Observe que no caso de seu Conector de saída, não é possível executar a Descoberta de Esquema porque não há nenhum arquivo `Output.xml` para ser descoberto.

18. Para usuários familiarizados com a versão 6.x e anterior, também é possível usar a sintaxe pré-7.0:

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

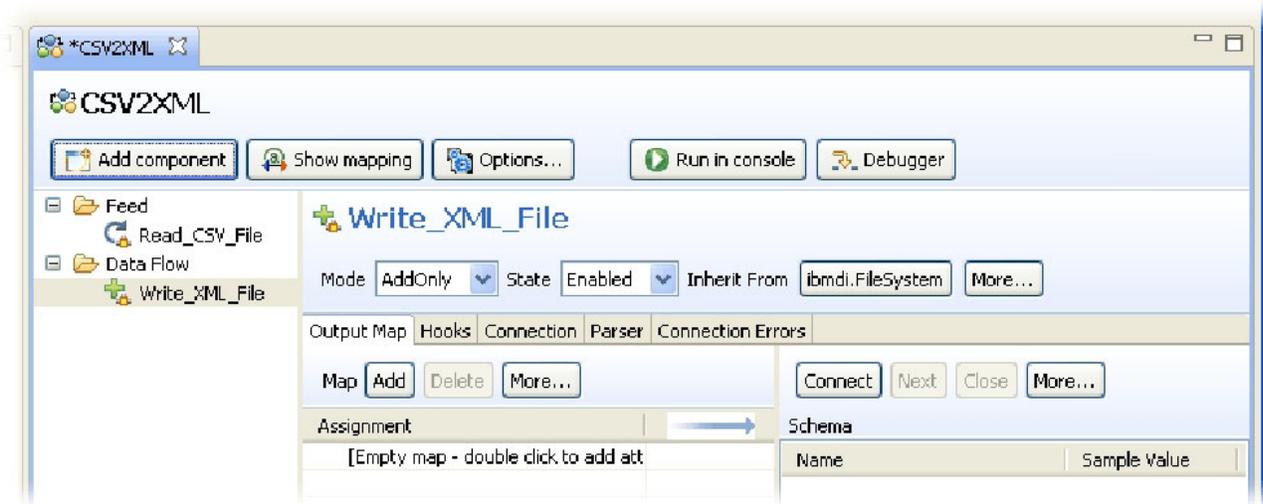


Figura 22. AL com Dois Conectores no Local

Você pode ter observado que ao selecionar um componente, seus detalhes aparecem na parte direita da tela do editor. Sempre que você seleciona a pasta 'Alimentação' ou 'Fluxo de Dados', é apresentada uma visão geral de todos os **Mapas de Atributos** para esse AssemblyLine. Essa é uma exibição conveniente para copiar seus Atributos de entrada para o Mapa de Saída de seu Conector mais recente. Portanto, exiba essa tela agora clicando em 'Alimentação' ou 'Fluxo de Dados'.

Aqui, você vê a lista de Atributos (três no total) que estão sendo trazidos para seu AL pelo Conector no modo Agente Iterativo. Selecione estes Atributos do Mapa de Entrada¹⁹ e arraste-os para baixo até o Mapa de Saída do Conector 'Write_XML_File', completando o fluxo de dados.

19. Você pode usar Control+clique para selecionar vários ou usar Shift+clique para selecionar um intervalo.

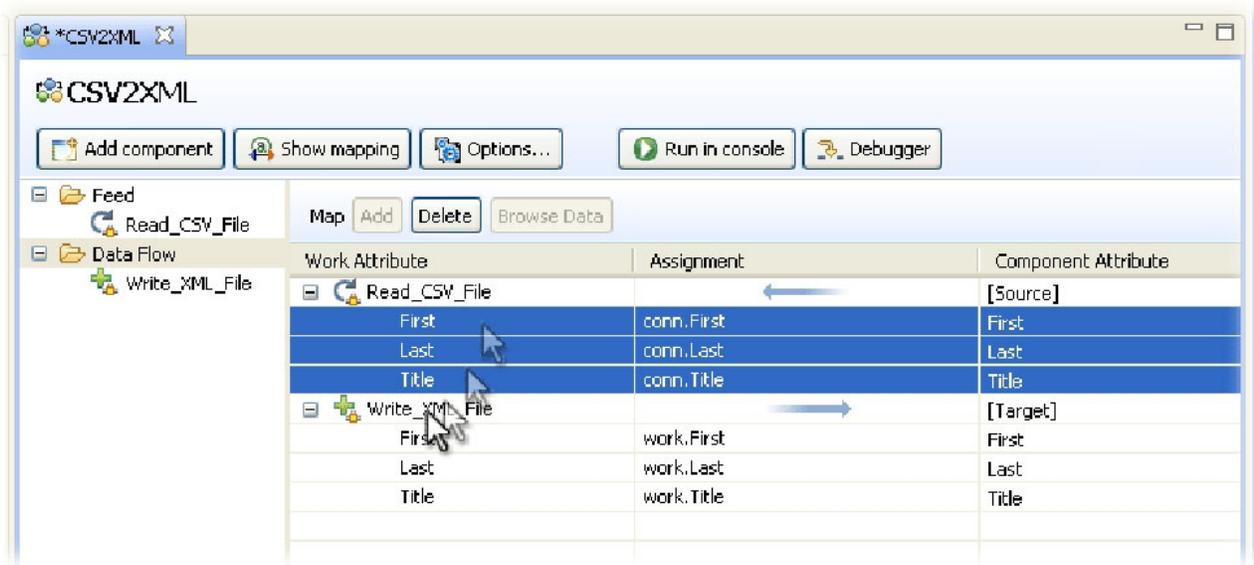


Figura 23. Arrastando Atributos para o Mapa de Saída

Observe como a Designação é convertida automaticamente do formato de entrada para saída. Por exemplo, o primeiro item do mapa no Mapa de Entrada de seu Conector 'Read_CSV_File' criará um atributo na Entrada de Trabalho nomeado 'First' para conter quaisquer valores localizados em conn.First (ou seja, o Atributo denominado 'First' que foi lido na Entrada Conn). Quando você arrasta essa regra de mapeamento de entrada para um Mapa de Saída, sua designação é alterada de forma que o valor agora seja proveniente da Entrada de Trabalho e esteja criando um Atributo de destino no cache do Conector (a Entrada Conn).

Se você deseja alterar a origem para qualquer regra de mapeamento, edite a designação. Para alterar o nome do Atributo para o qual está mapeando, simplesmente clique com o botão direito do mouse nele e renomeie-o. Faça isso agora para as duas primeiras regras do Mapa de Saída.

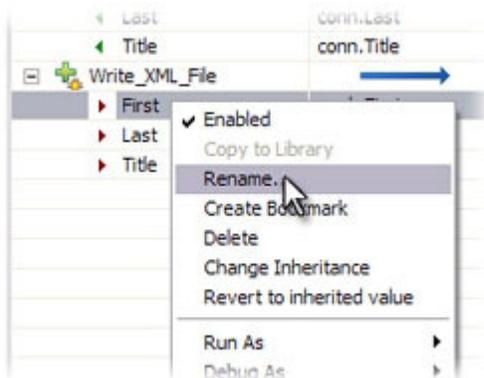


Figura 24. Renomeando uma Regra do Mapa de Atributos

Altere 'First' para 'FirstName' e 'Last' para 'LastName'.

Agora inclua um novo item de mapa nesse Mapa de Saída, clicando com o botão direito do mouse no próprio Mapa de Saída 'Write_XML_File' e escolhendo **Incluir**

Atributo.

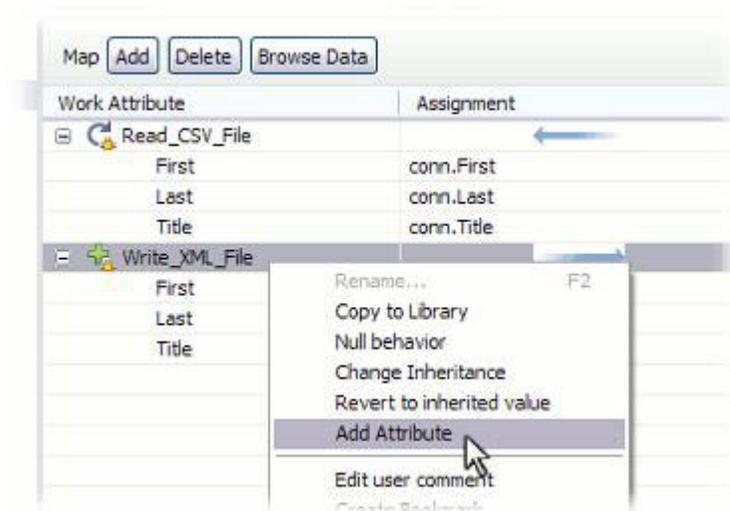


Figura 25. Incluindo o Atributo 'FullName' no Mapa de Saída

Chame o destino dessa nova regra de mapeamento 'FullName', pressione **OK** e, em seguida, clique duas vezes nele para editar sua designação. Isso abre o painel do editor de Script e apresenta um script de designação padrão: `work.FullName`. Naturalmente, não há nenhum Atributo 'FullName' na Entrada de Trabalho, portanto esse mapa não poderá configurar nenhum valor. Em vez disso, você deve *computar* esse valor alterando o script de forma que concatene os Atributos `First` e `Last`, deixando um único espaço entre esses valores:

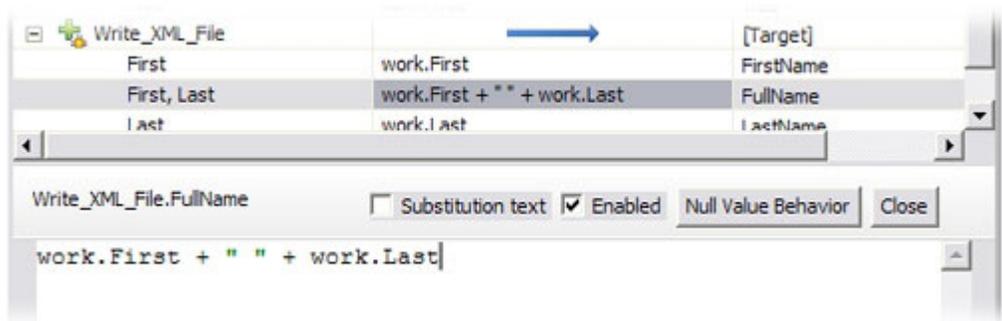


Figura 26. Editando a Designação

O script deverá ser o seguinte:

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

Observe que nenhum ponto e vírgula de terminação é necessário para scripts de designação do Mapa de Atributos de linha única como esta²⁰. Quando concluir, pressione o botão **Fechar** na parte superior direita do painel do editor de Script.

Executando seu Assemblyline

Agora é o momento de testar o AL.

20. A sintaxe pré-7.0 também é suportada, portanto os scripts de designação do mapa ainda podem iniciar com `ret.value =`.

Faça isso pressionando o botão **Executar** na parte superior do Editor de AssemblyLine.



Figura 27. O Botão Executar

Agora, você verá uma nova guia aberta com uma *Janela Executar* mostrando a saída do log proveniente de seu AssemblyLine.

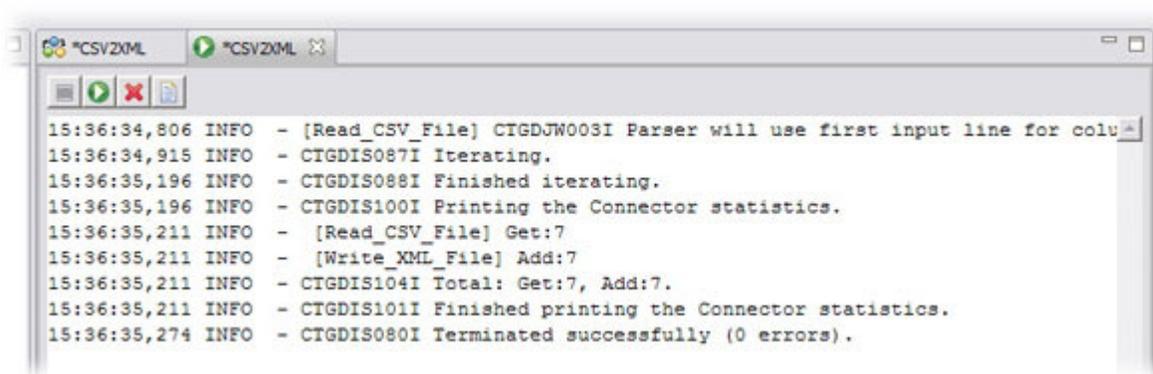


Figura 28. Saída do Log a partir da Execução do AssemblyLine

O CE usou na realidade o AssemblyLine com todos os seus componentes e exportou uma *Configuração* – um documento XML que define o trabalho designado a um Servidor IBM Security Directory Integrator. Depois, ele canalizou essa Configuração para o Servidor e o instruiu a executar seu AL, capturando toda a saída do log para exibição na tela.



Figura 29. Barra de Botões para a Janela Saída do Log

A janela Executar inclui uma barra de botões com opções para parar o AL, reiniciá-lo depois que ele for parado e para limpar o conteúdo do log. O botão à direita abre a saída do log atual em um editor externo.

A saída do log de um AssemblyLine termina com as estatísticas de todos os componentes envolvidos, que em seu caso são apenas dois Conectores. A partir das informações acima, fica claro que sete entradas foram lidas a partir do arquivo CSV e sete nós foram gravados no documento XML.

Você pode localizar seu arquivo de saída no disco e abri-lo em uma janela do navegador para confirmar seus resultados. Você também pode usar o Navegador de Dados, clicando com o botão direito do mouse em seu Conector de saída (Write_XML_File) e selecionando **Pesquisar Dados**.

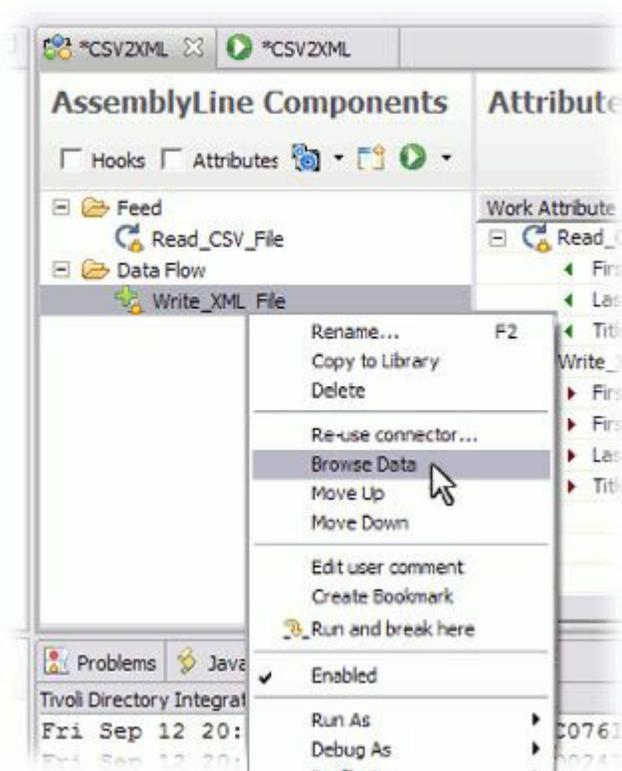


Figura 30. Pesquisando Dados Criados por um Conector de Saída

Isso exibe o Navegador de Dados para o Conector escolhido, configurado e pronto para prosseguir. Pressione o botão **Conectar** e, em seguida, **Próximo** para ler e exibir os dados de saída.

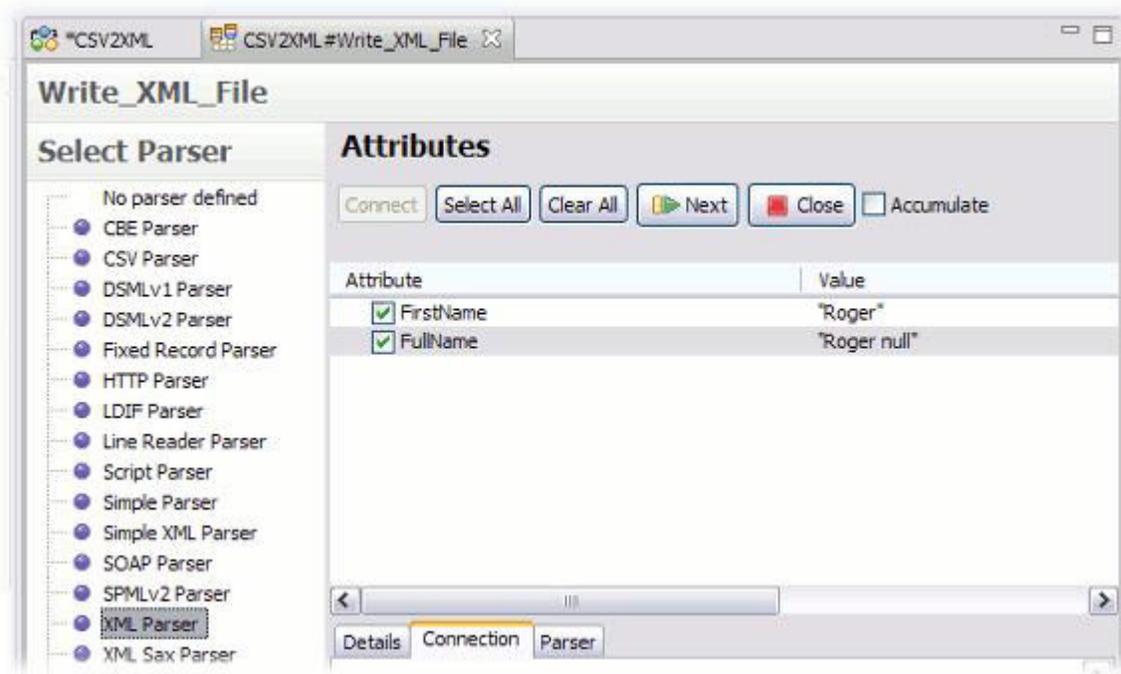


Figura 31. Pesquisando o XML Resultante

O XML de saída deveria ser uma representação exata dos valores de entrada, além de sua lógica de mapeamento; tudo parece correto, exceto a quarta entrada (Roger), que não contém o 'LastName' e o 'Title', e possui este valor de 'FullName' calculado: Roger null.

Se examinar os dados de entrada minuciosamente, você verá que uma das linhas 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

```

Dados de entrada ausentes e inválidos são um acontecimento comum; sua solução precisará ser preparada para filtrar ou corrigir isso durante o processamento.

Comportamento Nulo: Lidando com Atributos/Valores Ausentes

Para lidar com valores ausentes, é possível usar o recurso Comportamento Nulo integrado de Mapas de Atributos ou você mesmo pode detectá-lo e tratá-lo.

Vamos iniciar configurando o Comportamento Nulo. Faça isto clicando com o botão direito do mouse no Conector 'Read_CSV_File' no painel Mapas de Atributos (não na visualização em árvore de componentes do AL) e selecionando

Comportamento Nulo a partir do menu de contexto²¹.

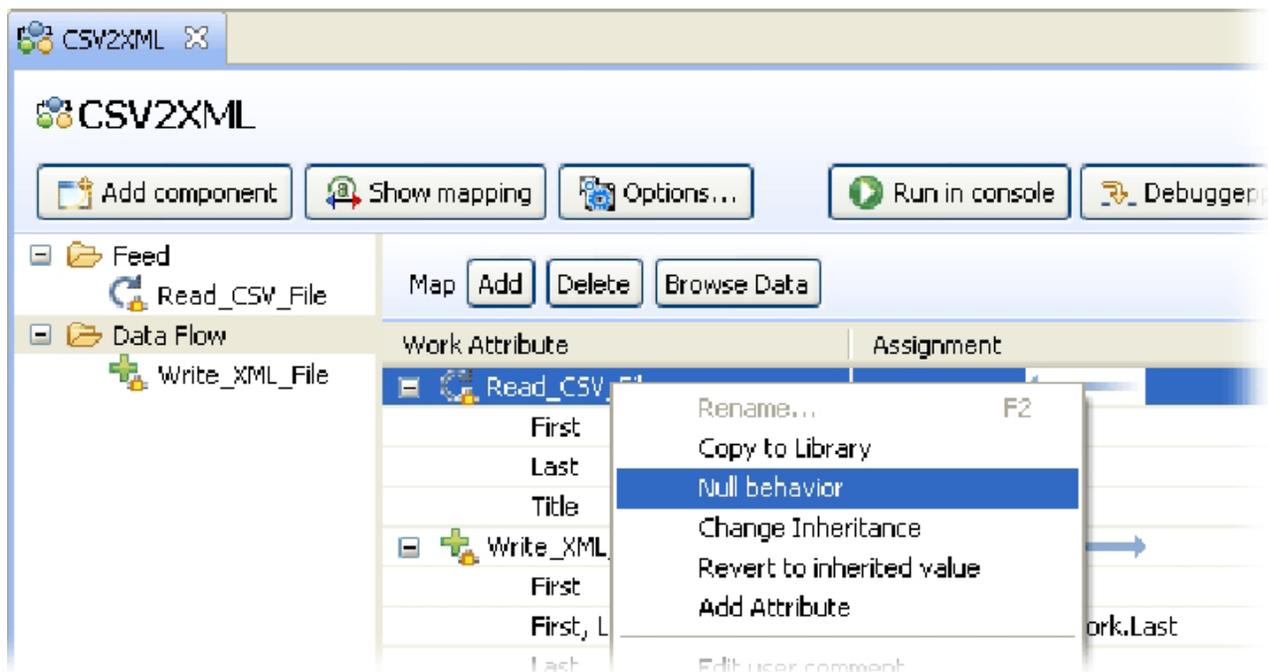


Figura 32. Botão Comportamento Nulo para Configuração no Nível do AL

Isso exibe o diálogo Comportamento Nulo, onde é possível configurar como *nulo* está definido – que pode variar dependendo do tipo de origem a partir do qual você está lendo – e também como essa situação deve ser tratada. Por padrão, *nulo* significa que um Atributo está ausente e a manipulação padrão é remover esse Atributo da operação de mapeamento. O resultado final disso é que nenhum Atributo com o nome especificado será localizado na Entrada de recebimento.

Quando estiver no diálogo Comportamento Nulo, use os botões de opções à direita para definir *nulo* como um valor de cadeia vazia e, então, aqueles à esquerda para especificar um valor-padrão de '* ausente *' a ser retornado neste caso.

21. Ou você pode selecionar o próprio Conector na lista de componentes do AL; pressione o botão **Mais...** na parte superior do Mapa de Entrada e escolha 'Comportamento Nulo'.

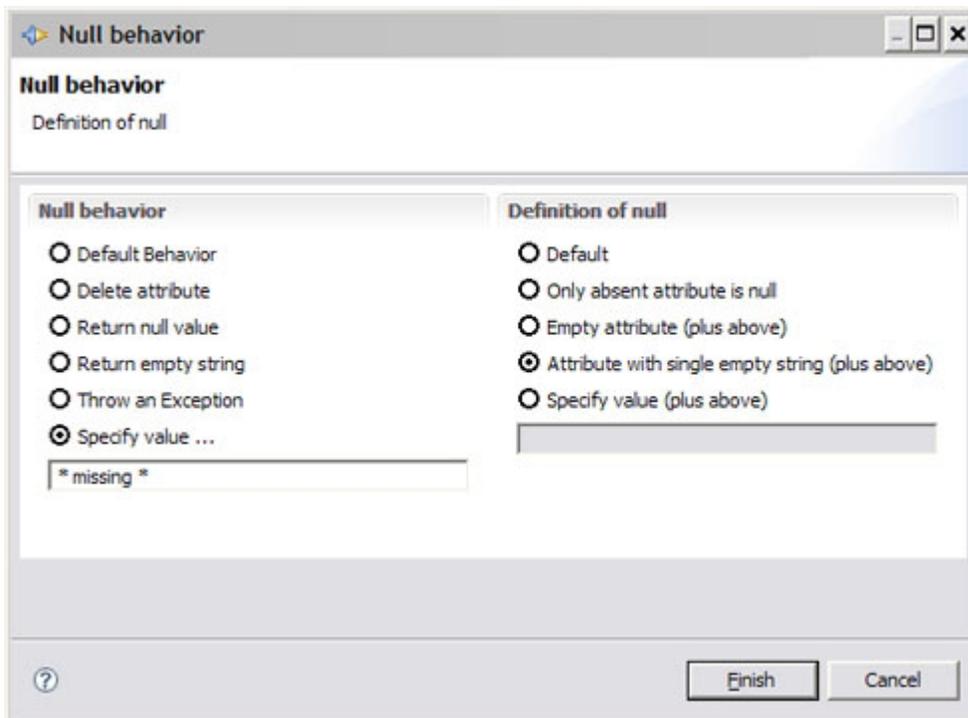


Figura 33. Diálogo de Configuração do Comportamento Nulo

Execute novamente seu AssemblyLine e atualize a janela do navegador exibindo o conteúdo de Output.xml. A entrada para 'Roger' deverá ter agora o valor *nulo* especial ("* ausente *") para 'Title' e 'LastName'.

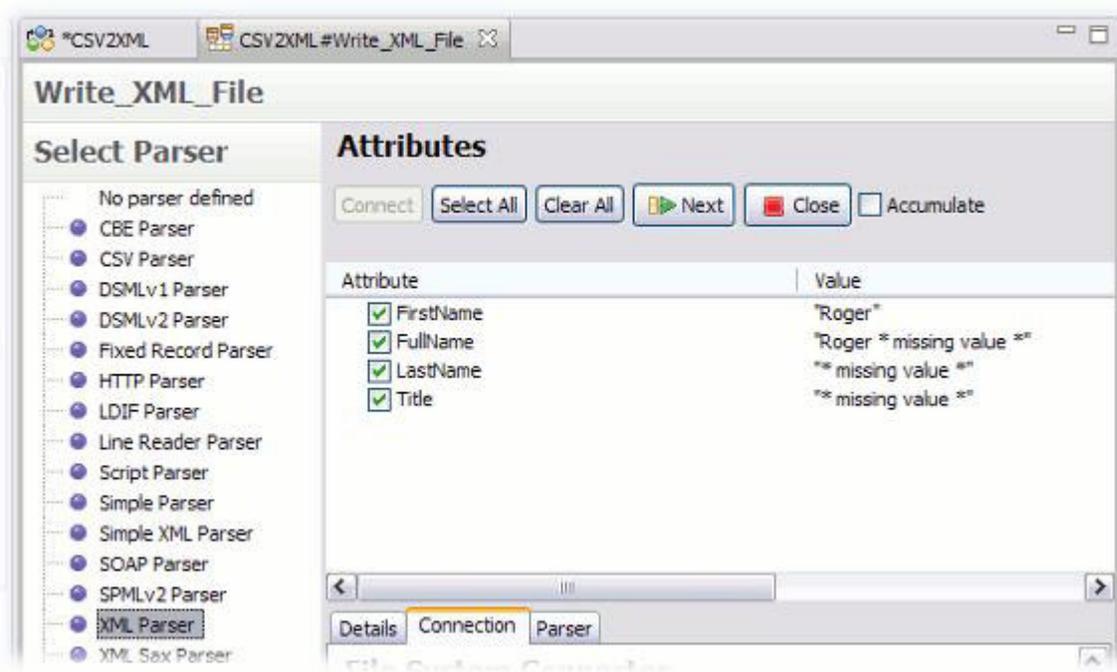


Figura 34. Resultado na Saída XML de Configurações do Comportamento Nulo

Isso está melhor – pelo menos é uma escolha consciente. Entretanto, às vezes, uma entrada está apenas muito incompleta para continuar o processamento. Em nosso cenário, são necessários pelo menos os valores para 'FirstName' e 'LastName' para computar 'FullName', portanto, agora você incluirá a lógica de filtragem em seu AssemblyLine para assegurar que todas as entradas preencham esse requisito.

Inicie clicando no botão **Inserir Componente** novamente e, em seguida, escolhendo **Componentes de Controle/Fluxo** nos botões de opções à esquerda. Depois, selecione 'IF' na lista, nomeie-o como 'Dados incompletos'²² e pressione **Concluir**.

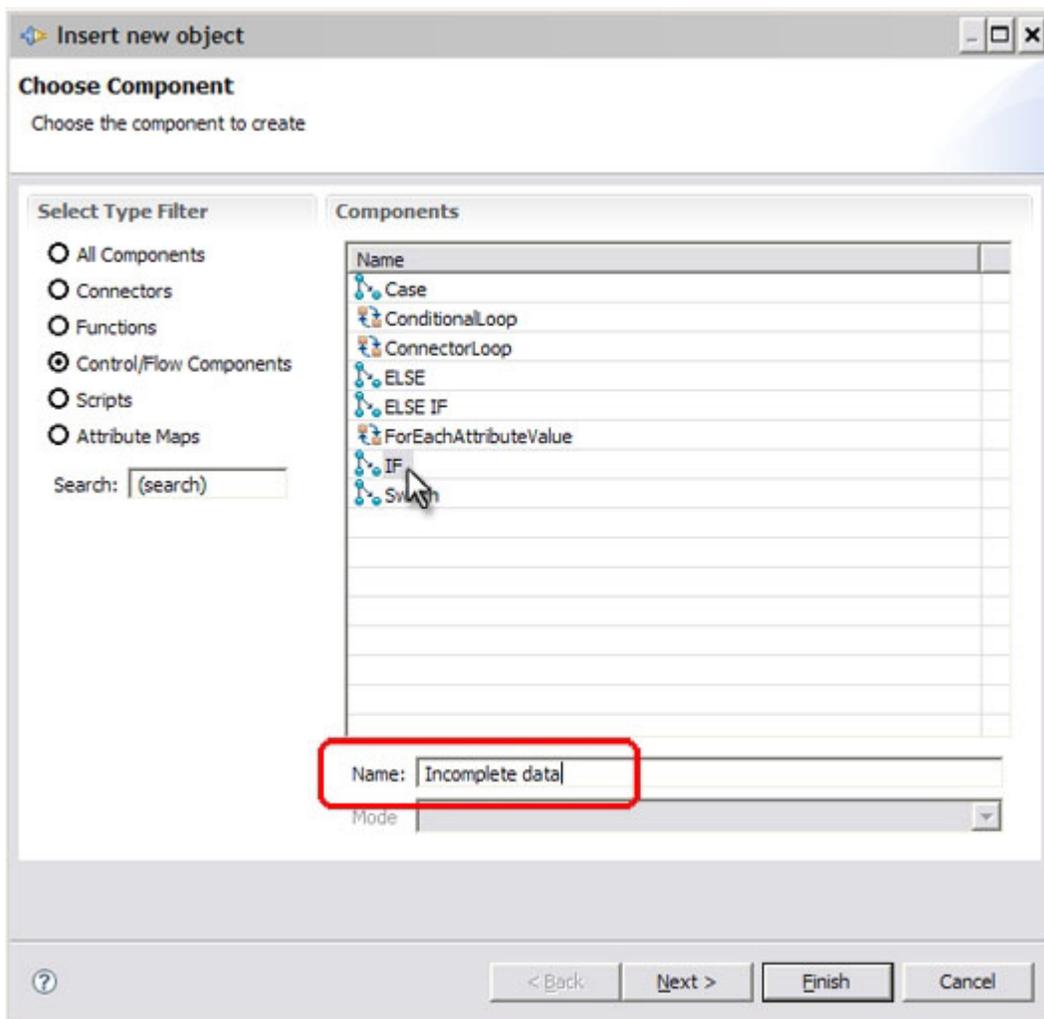


Figura 35. Selecionando o Componente de Ramificação IF

Agora, arraste essa ramificação IF acima de seu Conector de Saída. A área do editor de ramificação IF permite que você inclua suas condições.

22. Você foi aconselhado anteriormente a nomear Conectores da mesma forma que faria para uma variável de script. Isso se aplica também aos Function Components. Entretanto, isso é menos importante para os componentes do Mapa de Atributos, as Ramificações, os Loops e os Scripts, pois eles são acessados de forma aleatória diretamente a partir do script. Neste guia, foi dada uma prioridade maior para tornar o AssemblyLine fácil de ser lido.

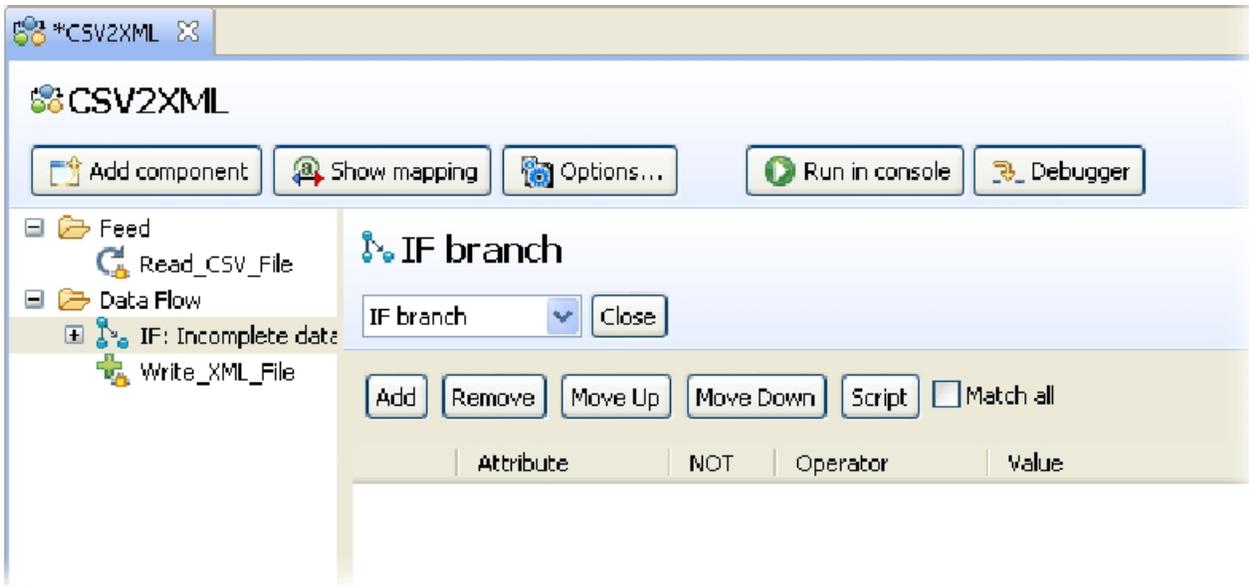


Figura 36. Editando Condições para a Ramificação IF

Aqui, você tem a opção de incluir condições simples ou escrever um trecho de Script – ou ambos. Observe a caixa de opção **Corresponder Todos**, que decide se suas Condições (simples e em script) são avaliadas com um *OU* implícito entre elas, quando desmarcada, ou com *E*, quando selecionada.

Inclua uma condição simples pressionando o botão **Incluir**. Em seguida, selecione o Atributo 'First' no menu suspenso à esquerda e, depois, o operador 'possui valor(es)'. Negue essa condição alternando o valor da coluna **Não**. Nada precisa ser especificado no campo à direita ao usar o operador 'possui valor(es)'. Agora, inclua também uma condição *não possui valores* similar para o Atributo nomeado 'Last'. Por último, certifique-se de que a caixa de opção **Corresponder Todos** esteja desmarcada – o que se deduz *Corresponder Qualquer Um* – para que a falta de valores para qualquer um dos Atributos acione essa ramificação.

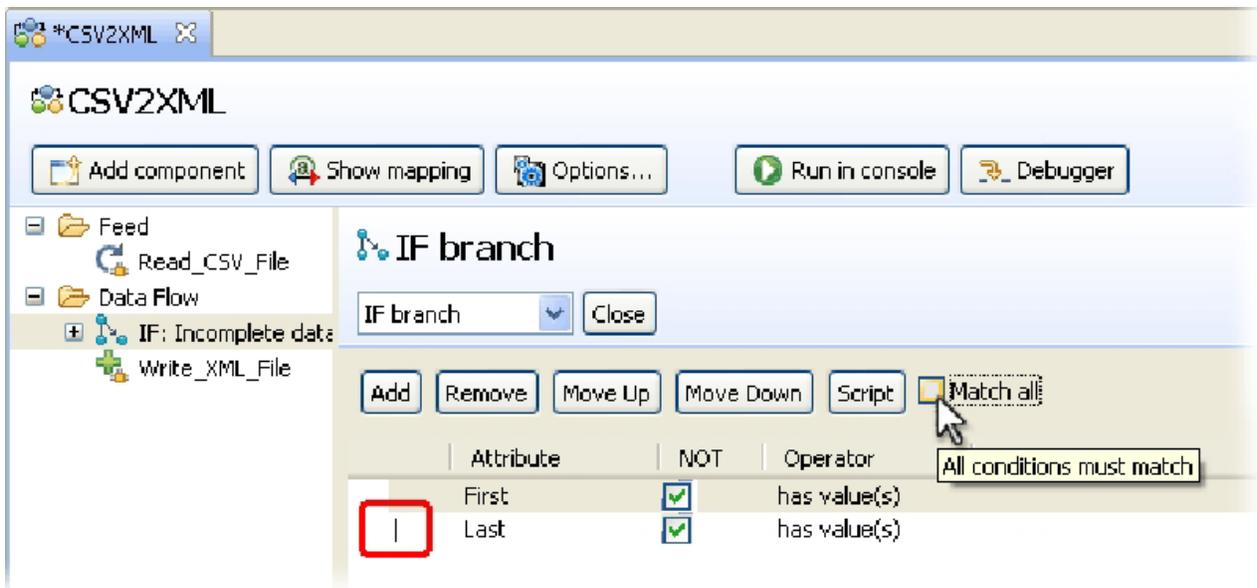


Figura 37. Incluindo Condições Simples para a Ramificação IF

Uma Ramificação IF redireciona o fluxo de execução do AssemblyLine sempre que as Condições especificadas são avaliadas como *verdadeiras*. Em seu caso, o processamento continuará para aqueles componentes colocados sob a ramificação se os Atributos 'First' ou 'Last' não tiverem valores designados a eles – ou mesmo se nenhum deles existir na Entrada de Trabalho. Ou seja, o operador 'possui valor(es)' também inclui a verificação para 'existe'.

Agora, expanda a ramificação IF e clique duas vezes no marcador exibido sob ela para inserir um componente. No diálogo **Escolher Componente**, clique no botão de opções para **Scripts**, selecione aquele chamado 'Script' e pressione **Concluir**. Renomeie esse componente de Script (também chamado 'SC') para 'Gravar no log' e, então, insira nele o seguinte fragmento de JavaScript²³:

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

A variável *task* usada aqui referencia o próprio AssemblyLine e fornece várias funções úteis, como o método `logmsg()`. Essa chamada em script faz com que a mensagem de texto especificada seja gravada em sua saída de log.

Para tornar a saída do log ainda mais informativa, vamos incluir também o conteúdo atual da Entrada de Trabalho. Faça isso clicando com o botão direito do mouse na ramificação IF, selecionando **Incluir Componente...** e escolhendo novamente o botão de opções para **Scripts**. Desta vez, selecione o componente de Script predefinido rotulado 'Dump de Entrada de Trabalho'.

Por último, você deve instruir o AL a parar o ciclo atual neste ponto e retornar o controle para o Conector de Agente Iterativo, para que ele possa ler a próxima linha CSV, filtrando eficazmente a Entrada atual da saída XML. A menos que você

23. O editor de Script fornece um recurso chamado *conclusão de código* que mostra quais opções você possui. Por exemplo, digite "tas" no editor de Script e, em seguida, pressione a combinação de teclas Ctrl + Espaço para abrir o menu suspenso de conclusão de código, que deverá fornecer uma única opção: `task`. Se você pressionar Enter, essa opção será selecionada e inserida em seu script. Agora, digite a tecla de ponto (.) para que seu script torne-se "task." e aguarde um momento; um novo menu suspenso de conclusão de código aparecerá, desta vez com uma lista de todos os métodos e propriedades que podem ser acessados no objeto `task`.

mesmo especifique esse comportamento, o controle continuará com o primeiro componente após a ramificação IF. Portanto, mais uma vez, você deve clicar com o botão direito do mouse em sua ramificação IF e, em seguida, em **Incluir Componente...** Escolha **Scripts** e, em seguida, selecione o SC chamado 'Fluxo de Saída'. Agora, seu AL deverá ser semelhante a este:

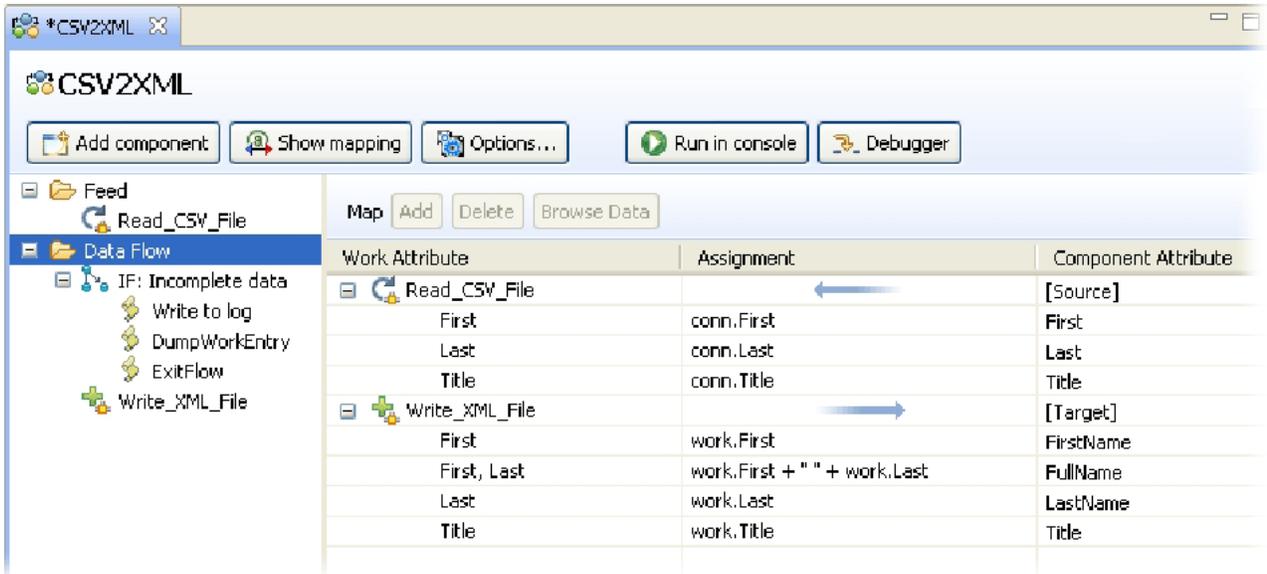


Figura 38. Seu Primeiro AssemblyLine Concluído

Antes de executar seu AssemblyLine novamente, você precisará abrir o diálogo Comportamento Nulo mais uma vez e restaurar as seleções padrão de definição e de comportamento – caso contrário, suas condições da Ramificação IF nunca serão avaliadas como *true*.

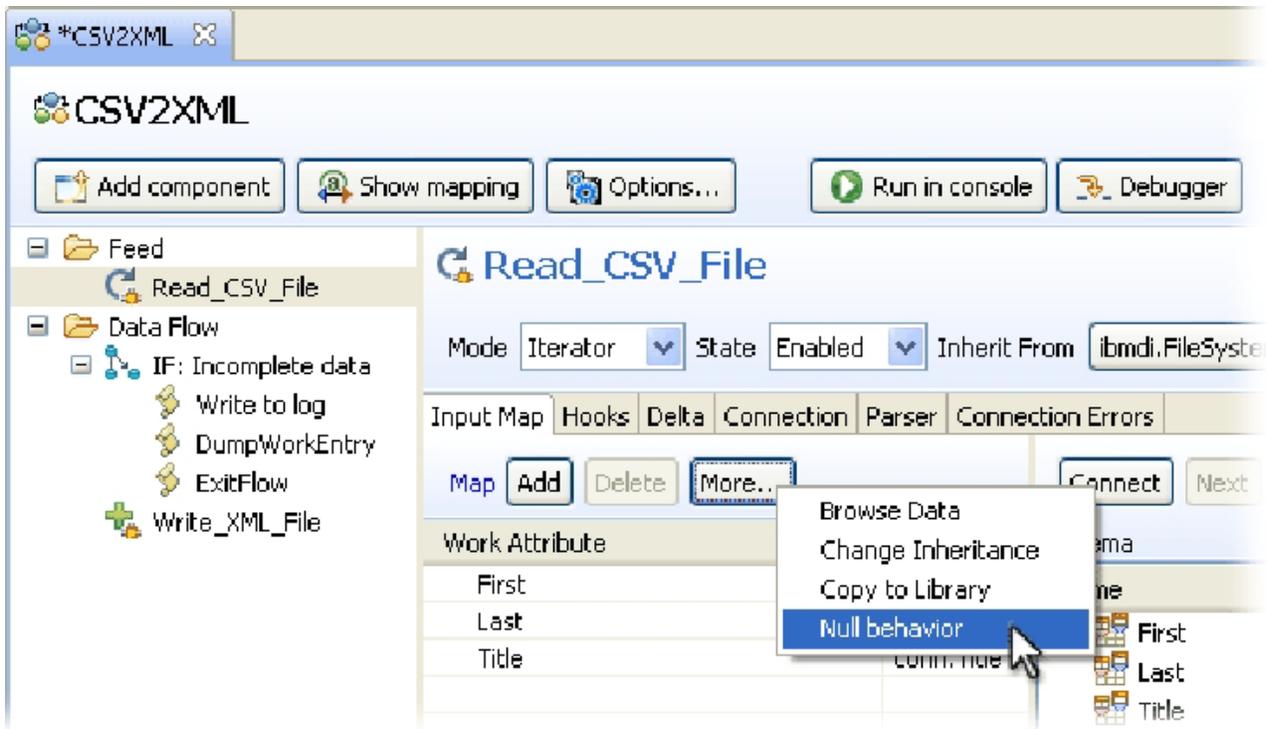


Figura 39. Reconfigurando o Comportamento Nulo para o AssemblyLine

Agora, quando você executar seu AssemblyLine novamente, verá sua mensagem seguida pelo dump de Entrada de Trabalho:

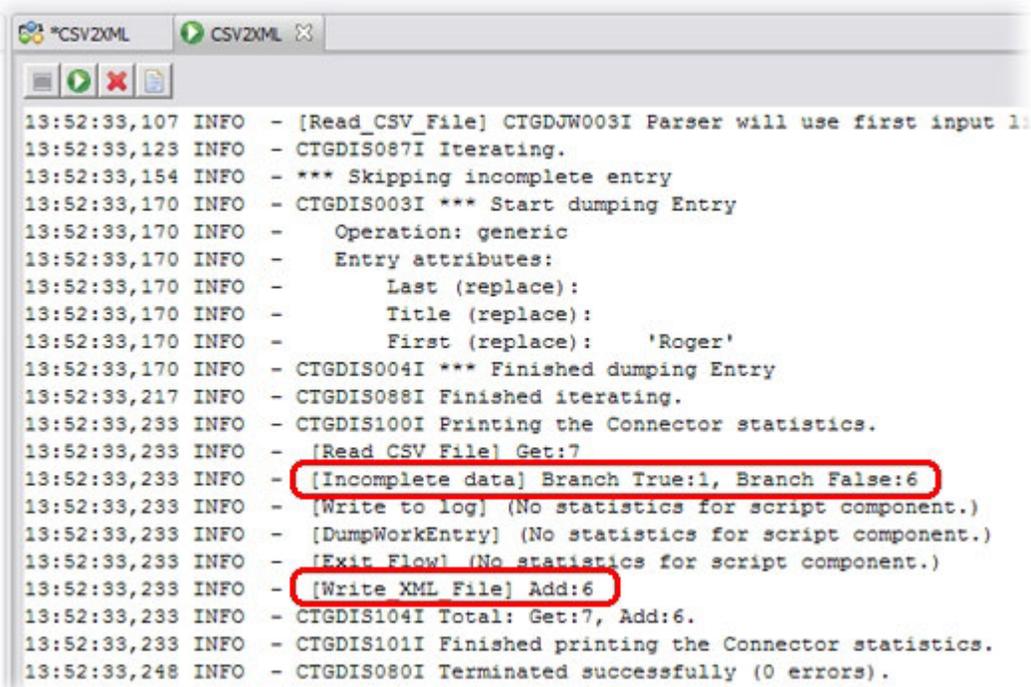


Figura 40. Saída do Log com suas Mensagens e o Dump de Entrada de Trabalho

A partir das estatísticas, você pode ver que sua ramificação IF 'Dados incompletos' foi verdadeira uma vez, resultando na inclusão de apenas seis nós em sua saída XML. Se você atualizar novamente a janela Navegador de Dados, então verá que, de fato, 'Roger' sumiu.

Depurando seu AssemblyLine

Um dos recursos mais eficazes do IBM Security Directory Integrator é seu Depurador do AssemblyLine integrado, que permite acompanhar a execução de seu AL, visualizar e até mesmo modificar os dados em andamento.

Vamos avançar passo a passo em seu primeiro AssemblyLine, clicando no botão de sessão **Iniciar Depuração**.

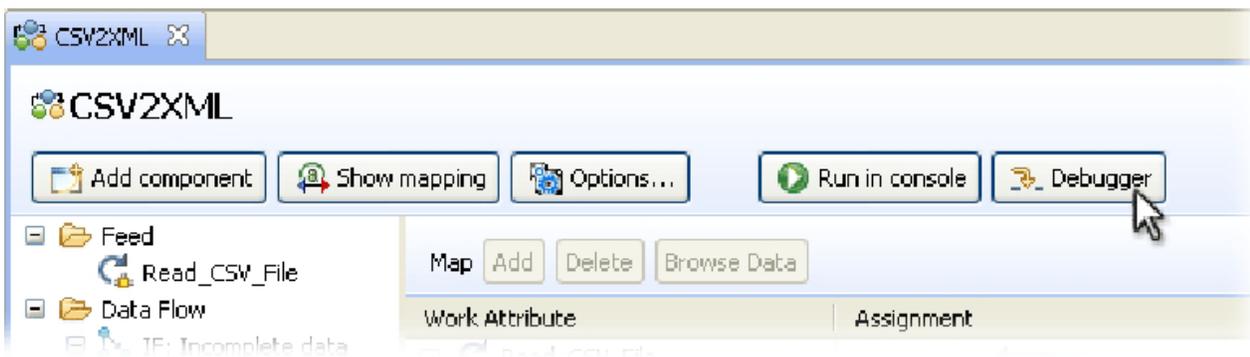


Figura 41. Depurando seu AssemblyLine

Em vez da janela Executar Console padrão, você entrará no Depurador.

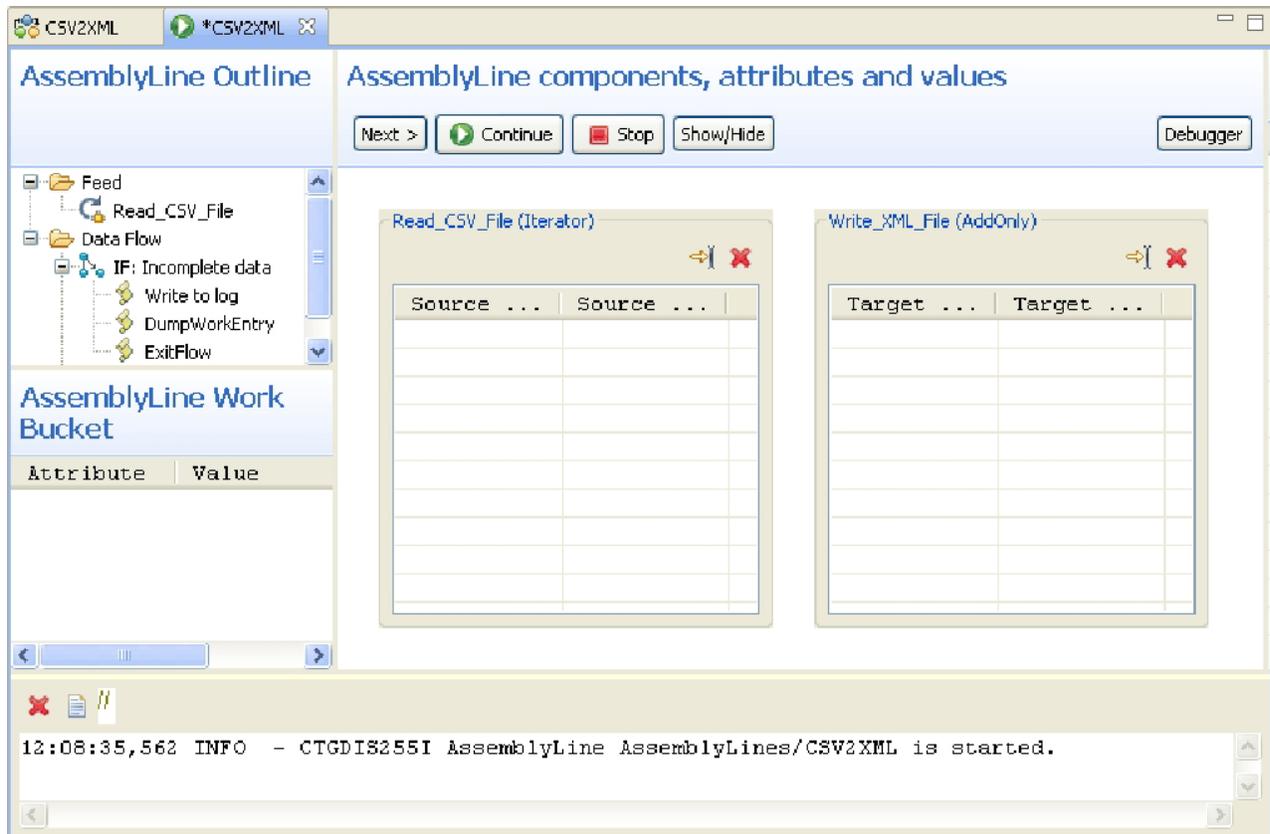


Figura 42. O Data Stepper do AssemblyLine

O Depurador do AssemblyLine oferece dois modos: o Data Stepper, que fornece recursos de teste simples e diretos, e o Depurador do AL avançado, em que você pode obter informações mais detalhadas - como avançar passo a passo em scripts, trabalhando interativamente com bibliotecas Java e modificando dados no arquivo.

O Data Stepper é uma ferramenta útil para avançar passo a passo na execução dos Conectores do AssemblyLine e para visualizar os dados lidos, gravados e transformados. Esta tela é dividida em três áreas principais:

- A **Estrutura de Tópicos do AssemblyLine** mostra seu AL, realçando onde a execução é pausada;
- O **Depósito de Trabalho do AssemblyLine**, que exhibe todos os Atributos mapeados para a Entrada de Trabalho – isto é, aqueles encontrados nos componentes Mapas de Entrada ou Mapa de Atributos.
- **Componentes do AssemblyLine, Atributos e Valores**, em que você tem uma linha de botões para controlar a sessão de depuração e um conjunto de grades de exibição de dados para todos os Conectores no AL.

Ao longo da parte inferior, você pode ver uma janela de saída do Console que mostra as mesmas informações obtidas quando executou seu AssemblyLine usando o botão Executar.

Neste ponto, seu AssemblyLine foi despachado para o Server de teste e está pronto para iniciar a execução em seu comando. Pressione o botão Avançar para iniciar as etapas.

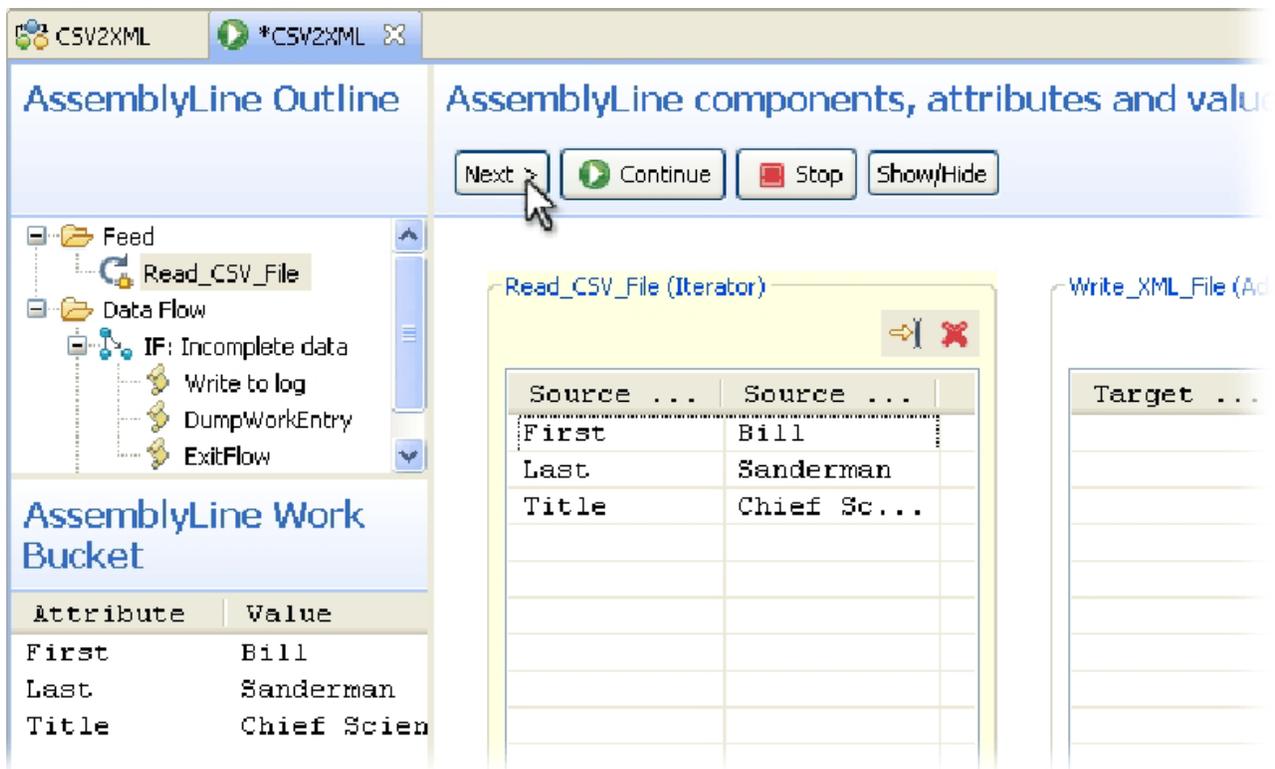


Figura 43. Avançando na Execução do AL

Observe como três situações ocorrem na tela: a **Estrutura de Tópicos do AssemblyLine** mostra que o Conector "Read_CSV_File" está ativo atualmente; o **Depósito de Trabalho do AssemblyLine** exibe os Atributos que acabaram de ser lidos por esse Conector; e a grade de exibição de dados desse Conector também é preenchida com esses Atributos. Cada vez que você pressiona o botão **Avançar** a execução continua com a etapa e as exibições de informações são atualizadas.

Também é possível usar o botão **Executar até aqui** na parte superior da grade de dados para que um Conector vá para este ponto. Faça isso agora para o Conector "Write_XML_File".

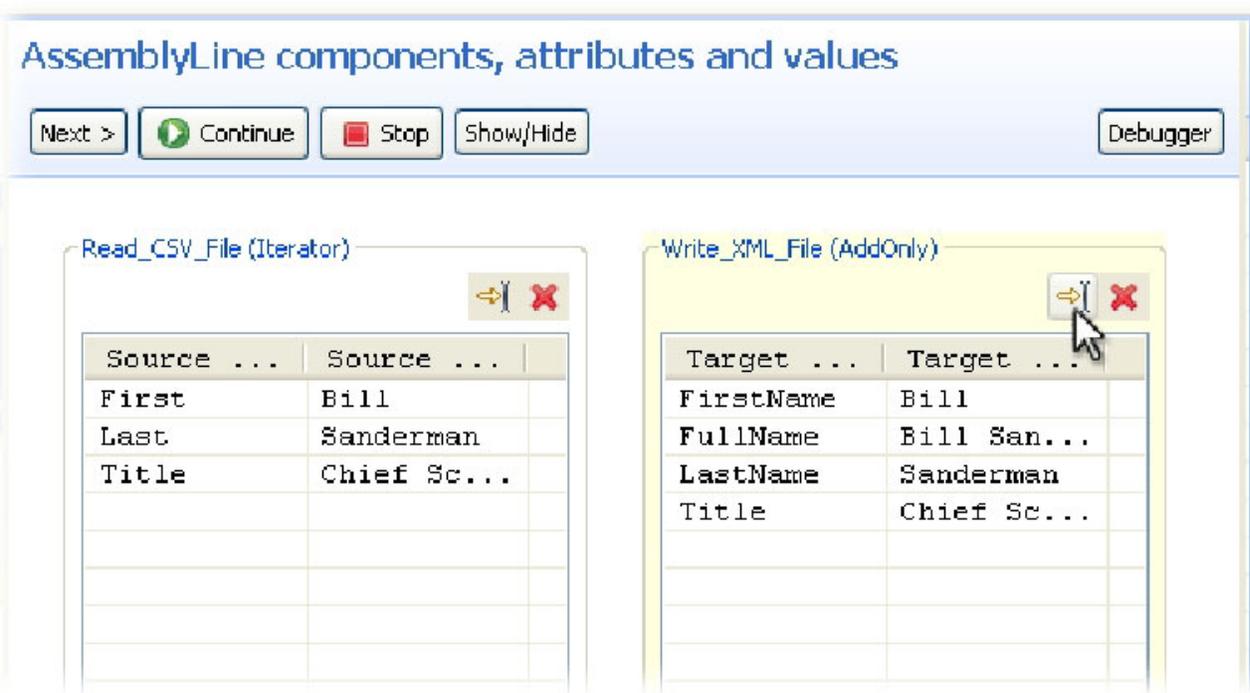


Figura 44. Avançando para o Conector Write_XML_File

Em seguida, essa grade de dados exibe os Atributos no Mapa de Saída desse Conector juntamente com seus valores – incluindo o valor computador para "FullName".

Agora, vamos analisar os botões da barra de ferramentas do Data Stepper para ver as opções que você tem:

- **Avançar >** move o processamento para o próximo componente e atualiza todas as áreas de exibição de dados;
- **Continuar** fará com que seu AL continue a execução até a conclusão;
- **Parar** termina a execução imediatamente;
- **Mostrar/Ocultar** permite decidir quais grades de dados do Conector são exibidas;
- **Depurador** alterna para o modo Depurador Completo, em que você pode avançar pelo código de script, configurar pontos de interrupção, visualizar e modificar quaisquer Atributos e variáveis de script e executar interativamente comandos de JavaScript no contexto de seu AssemblyLine em execução.

Embora o Data Stepper forneça uma grande quantidade de informações sobre como o AssemblyLine será executado, às vezes você precisa do poder incluído do Depurador avançado. Observe que é possível alternar entre os modos Passo a Passo e Avançado quando você desejar durante uma sessão de depuração. Teste-o agora pressionando o botão **Depurador**, localizado na extrema direita da linha de botões do Data Stepper.

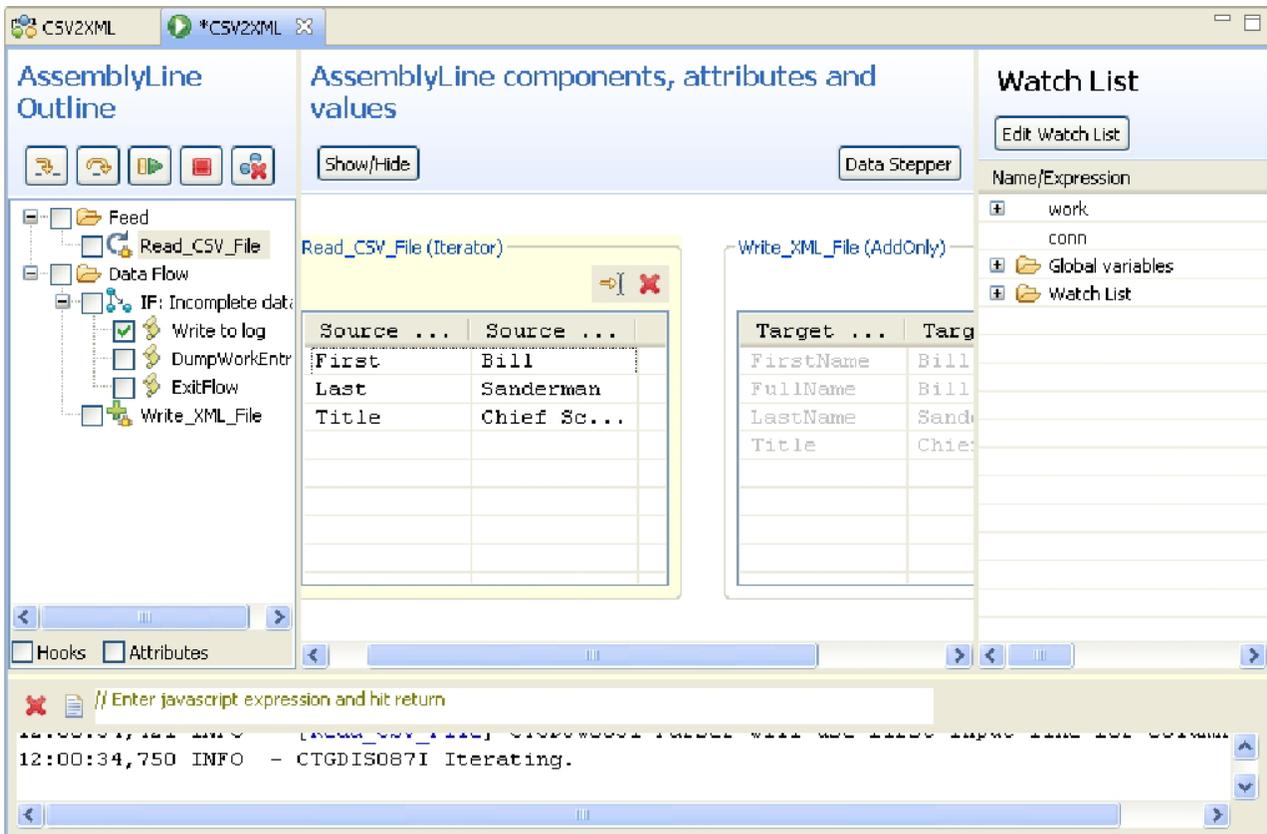


Figura 45. Modo Depurador Avançado

Quando você alterna os modos, a tela é redesenhada para fornecer novos controles para a **Estrutura de Tópicos do AssemblyLine** e o **Depósito de Trabalho do AssemblyLine** é substituído pela **Lista de Inspeção** no lado direito da janela. A **Lista de Inspeção** mostra os depósitos de Atributo padrão: work e conn. Também existe uma pasta chamada "Variáveis Globais" que, se aberta, exibirá todas as variáveis definidas para o seu AssemblyLine: as variáveis integradas, como **work** e **system**, e mais qualquer uma que você defina em seu código de script. A última pasta Inspeção é apenas para uso próprio e você pode incluir variáveis ou expressões JavaScript inteiras que deseje inspecionar usando o botão **Editar Lista de Inspeção**, na parte superior deste painel.

Voltando a sua atenção para a Estrutura de Tópicos do **AssemblyLine**, as caixas ao lado dos componentes nesta visualização em árvore são chamadas Pontos de Interrupção e é possível solicitar ao IBM Security Directory Integrator para pausar em qualquer componente durante a execução clicando em um deles. Também é possível clicar com o botão direito do mouse em qualquer nó existente e selecionar **Executar e Interromper Aqui** para que a execução do AL seja ativada neste ponto. A barra de ferramentas acima da estrutura de tópicos fornece alguns dos mesmos controles que você tinha no Data Stepper, além de mais alguns novos:



Figura 46. Botões do Depurador

Começando pelo lado esquerdo, estes botões são:

- **Avançar para** que permite avançar nos Mapas de Atributo, scripts e, até mesmo, nos fluxos de trabalho subjacentes do AssemblyLine e de seus componentes. Esses pontos de caminho nos fluxos integrados são chamados de "Ganchos" e são abordados em um exercício mais adiante;
- **Avançar** é o mesmo que o botão **Avançar >** visto no Data Stepper. No Depurador, ele também permite parar além de chamadas de função de script, em vez de ficar nelas;
- **Continuar** faz com que o AL seja executado até a conclusão (como no Data Stepper) ou até um Ponto de Interrupção ser atingido;
- **Parar** interrompe o AssemblyLine, como faz no Data Stepper;
- **Limpar Todos os Pontos de Interrupção** remove qualquer Ponto de Interrupção que você tenha configurado para o seu AL.

Para saber como os Pontos de Interrupção funcionam, tente configurar um para o script "Gravar no Log", clicando na caixa ao lado desse componente.

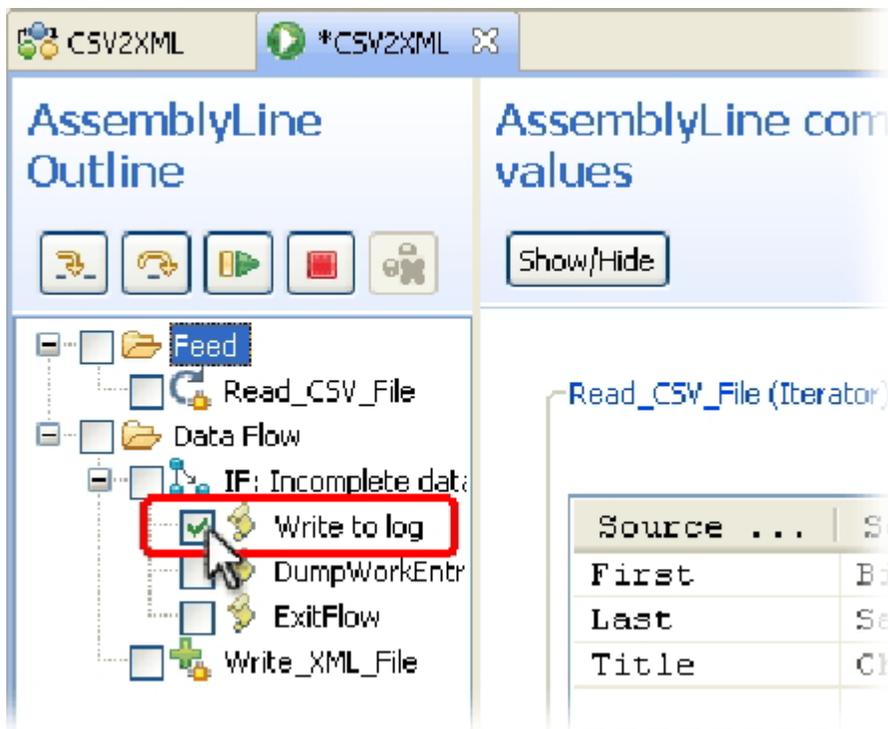


Figura 47. Configurando um Ponto de Interrupção

Agora, pressione o botão **Continuar** e o AL será executado até que IF-Branch seja definido como *true* e você localize o controle no Script "Gravar no Log". O IBM Security Directory Integrator também abre uma área de Script, permitindo avançar passo a passo pelo código aqui. Também é possível configurar Pontos de Interrupção em qualquer linha de script dando um clique duplo na margem à esquerda dessa linha.

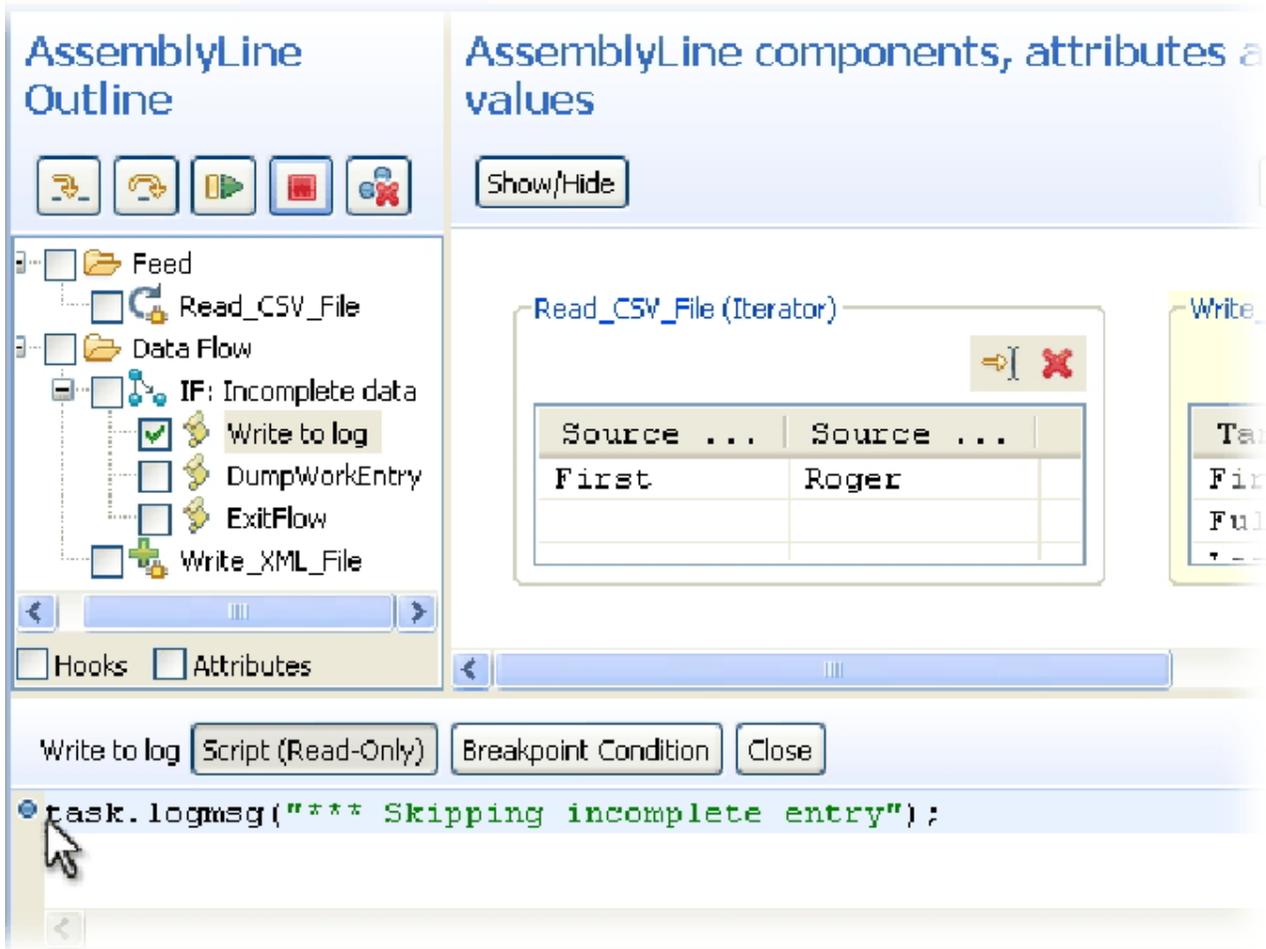


Figura 48. Configurando um Ponto de Interrupção no Script

Além disso, é possível dar um clique duplo em qualquer nó na lista de componentes para ativar a tela de Depuração. Como você pode ver na figura acima, há um botão com o intitulado **Condição do Ponto de Interrupção**. Você pode usar esse botão para configurar uma expressão de JavaScript que deve ser avaliada como *true* ou *false* e que determinará se um Ponto de Interrupção está ativo ou não. Por exemplo, o Ponto de Interrupção mostrado acima poderá ser configurado para *true* se:

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

ou

```
mycounter > 1000
```

Este recurso é muito útil para depurar problemas que ocorrem apenas em alguns conjuntos de dados de entrada.

E, se por algum motivo, você precisar voltar a uma etapa anterior, bastará parar e reiniciar a sua sessão de depuração. Também é possível retornar ao Data Stepper pressionando o botão Data Stepper.

Porém, antes de você sair do Depurador avançado, existe mais um recurso que é importante observar: a **linha de comandos de Avaliação de JavaScript**.

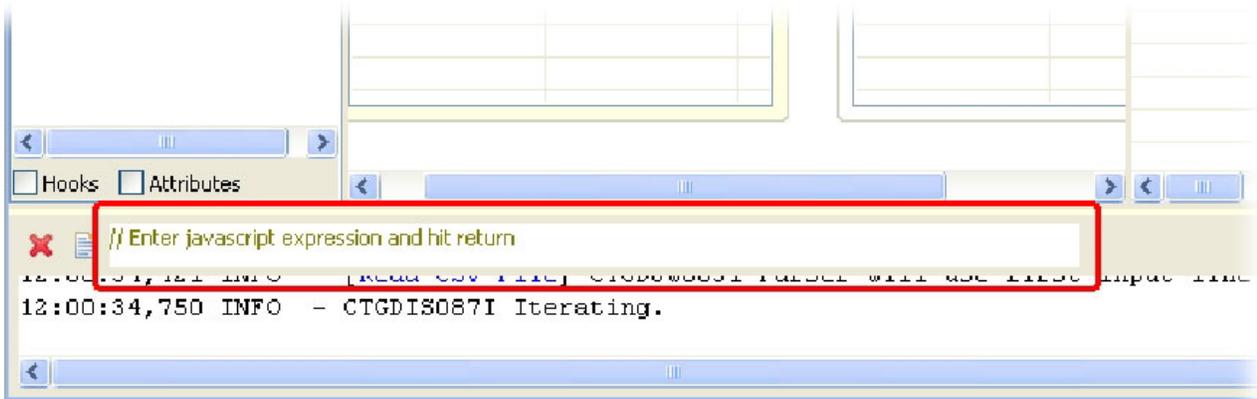


Figura 49. Linha de Comandos de Avaliação de JavaScript

Esse inocente campo de entrada de observação acima da área de saída de log permite executar qualquer fragmento de script no contexto de seu AssemblyLine em execução. Teste-o agora digitando este comando e pressionando Enter:

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

Isso exibirá o conteúdo da Entrada de Trabalho na janela de saída de log. Agora, tente isso:

```
i = 42
```

Você verá a seguinte mensagem aparecer no log: `i=42 >> 42.0`

Ela informa que você definiu uma nova variável ('i') com o valor 42. A expressão em si é avaliada (assim como fazem todas as instruções de script) para o valor da designação. Você também pode alterar os valores de variável e Atributos já definidos no seu AL, por exemplo:

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

Depois de executar esta linha, o valor ao Atributo "First" será "Rudy". O recurso para modificar dados sendo transferidos significa que você pode verificar se o seu AssemblyLine avança para toda a lógica de ramificação, permitindo que teste completamente a sua solução.

É altamente recomendável gastar algum tempo para familiarizar-se com o Data Stepper e o Depurador do AssemblyLine. Isso não apenas fornece um insight exclusivo sobre como o AL opera, incluindo todos os fluxos de trabalho integrados fornecidos pelo kernel do Servidor IBM Security Directory Integrator, mas também ajudará você a validar sua própria implementação e suposições sobre seus dados.

Consultando Dados de uma Origem Sequencial

Continuando com nosso cenário do tutorial, a próxima etapa é incluir a consulta a partir de D2.

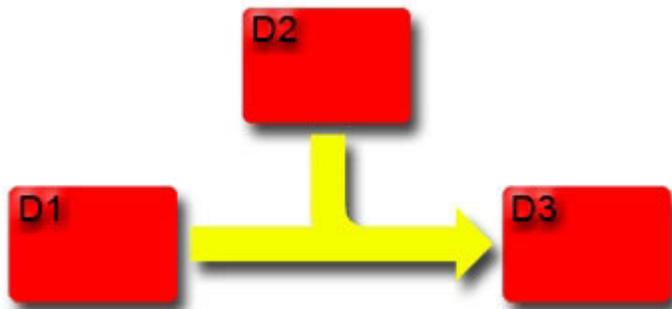


Figura 50. O Fluxograma do Cenário

O diretório Tutorial contém um arquivo denominado PhoneNumbers.xml que servirá como sua origem de dados D2. Esse arquivo contém uma série de entradas XML, cada uma delas com dois atributos: 'User' e 'telephoneNo'.

Sua tarefa será incluir 'telephoneNo' como parte dos dados gravados no documento XML de saída. Como você não pode acessar aleatoriamente esse arquivo de texto para executar uma Consulta, tal como faria para um banco de dados ou diretório, o número de telefone correto para cada entrada CSV será localizado efetuando loop no arquivo e comparando 'User' com o 'FullName' proveniente da entrada CSV atual.

Entretanto, 'FullName' está sendo computado primeiramente no Mapa de Saída do Conector 'Write_XML_File' – ou seja, é muito tarde para fazer a comparação. Isso significa que você deve mover esse Atributo computado do Mapa de Saída de 'Write_XML_File' para o Mapa de Entrada de 'Read_CSV_File'. Faça isso arrastando primeiro o item de Mapa de Atributos para cima, de um mapa para outro.

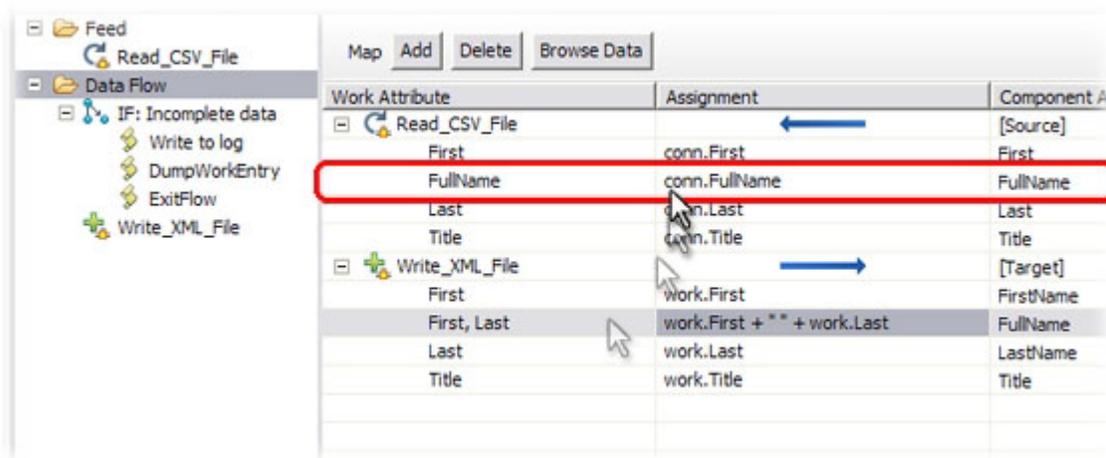


Figura 51. Arrastando 'FullName' para o Mapa de Entrada de seu Conector de Agente Iterativo

Agora, haverá um item de Mapa 'FullName' em ambos os mapas. Você precisa ajustar a designação do Mapa de Entrada, pois agora está mapeando da Entrada Conn para Work, e não ao contrário, como é o caso de um Mapa de Saída. Faça isso clicando duas vezes em 'FullName' sob 'Read_CSV_File' e alterando a designação para:

`conn.First + " " + conn.Last`

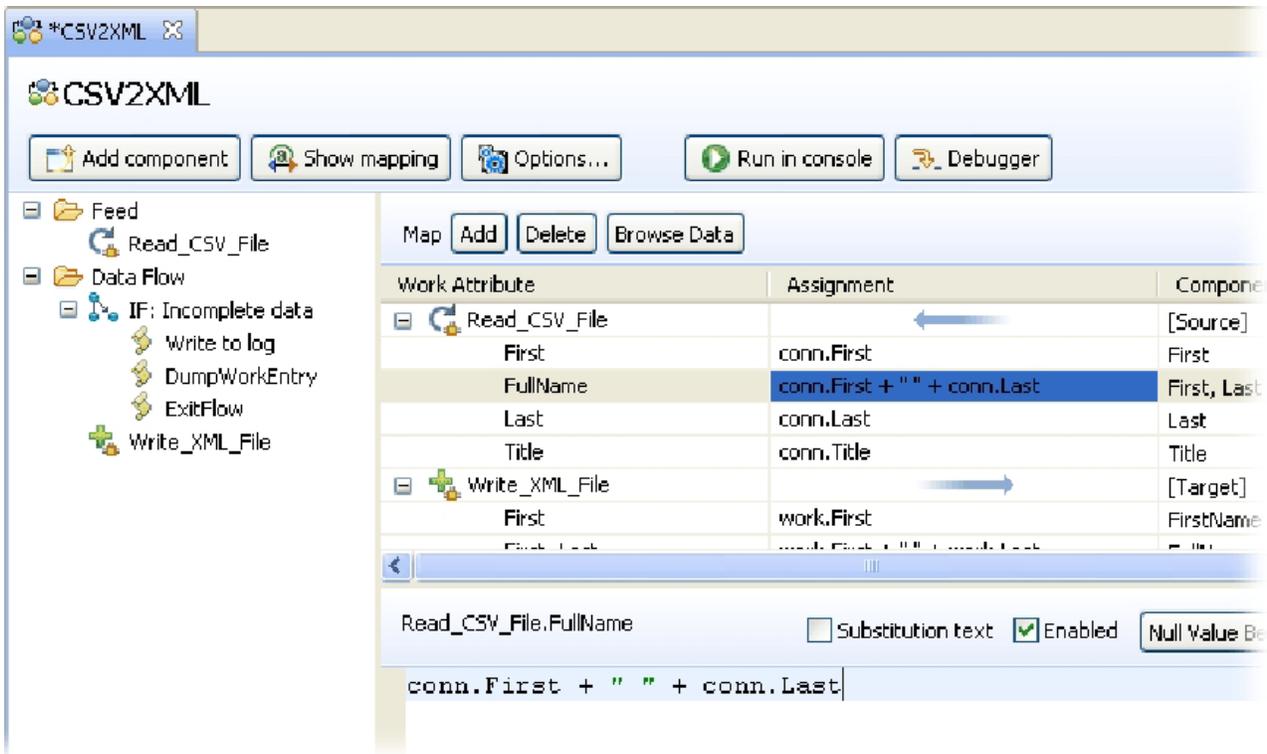


Figura 52. Editando a Designação para 'FullName'

Você também pode editar o item de Mapa de Saída 'FullName' para que sua designação seja simplesmente work.FullName, pois esse Atributo estará agora disponível na Entrada de Trabalho, graças a seu Mapa de Entrada modificado.

Agora, execute novamente seu AssemblyLine e verifique o Output.xml para certificar-se de que ele esteja inalterado. Depois de confirmado isso, você usará um componente *Loop* para ler através do arquivo PhoneNumbers.xml e procurará o número de cada usuário²⁴.

Inicie incluindo um novo componente na seção **Fluxo de Dados**, desta vez escolhendo o componente denominado *ConnectorLoop* e, em seguida, nomeando-o como 'TelephoneNumber'. Um *ConnectorLoop* é um componente em loop que usa um Conector para ler informações de uma origem de dados e, em seguida, efetua o ciclo de todos os componentes anexados a ele, uma vez para cada entrada retornada por esse Conector. Isso é semelhante ao comportamento *for-each* de um Conector de Agente Iterativo na seção Alimentação, que efetua o ciclo de componentes na seção **Fluxo de Dados** para cada entrada lida.

Arraste seu novo *ConnectorLoop* 'TelephoneNumber' entre a ramificação IF e o Conector 'Write_XML_File'. Certifique-se de que ele não acabe *dentro* da ramificação IF.

24. Há três tipos de componentes Loop: 1) O *ConnectorLoop*, que permite efetuar o ciclo nos dados retornados por um Conector no modo Agente Iterativo ou Consulta. Este é o tipo de Loop que você usará neste exercício; 2) o *ForEachAttributeValueLoop*, facilitando o loop através dos valores de um Atributo de multivalores, como aqueles que você localiza em sistemas como o Lotus Notes e em Diretórios LDAP; e 3) o *ConditionalLoop*, que usa Condições Simples e em script – exatamente como aquelas usadas por Ramificações – para controlar o ciclo.

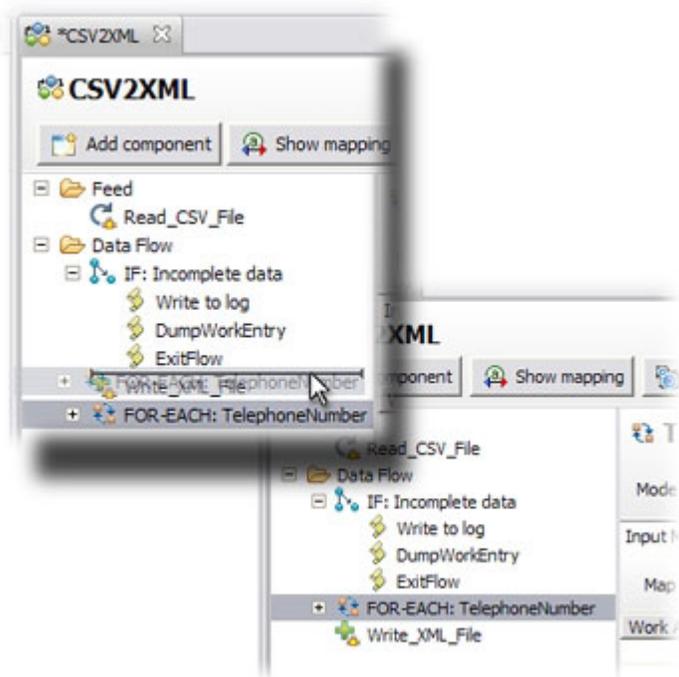


Figura 53. Arrastar o ConnectorLoop

Selecione-o agora para abrir seu editor, que é semelhante a um editor de Conector.

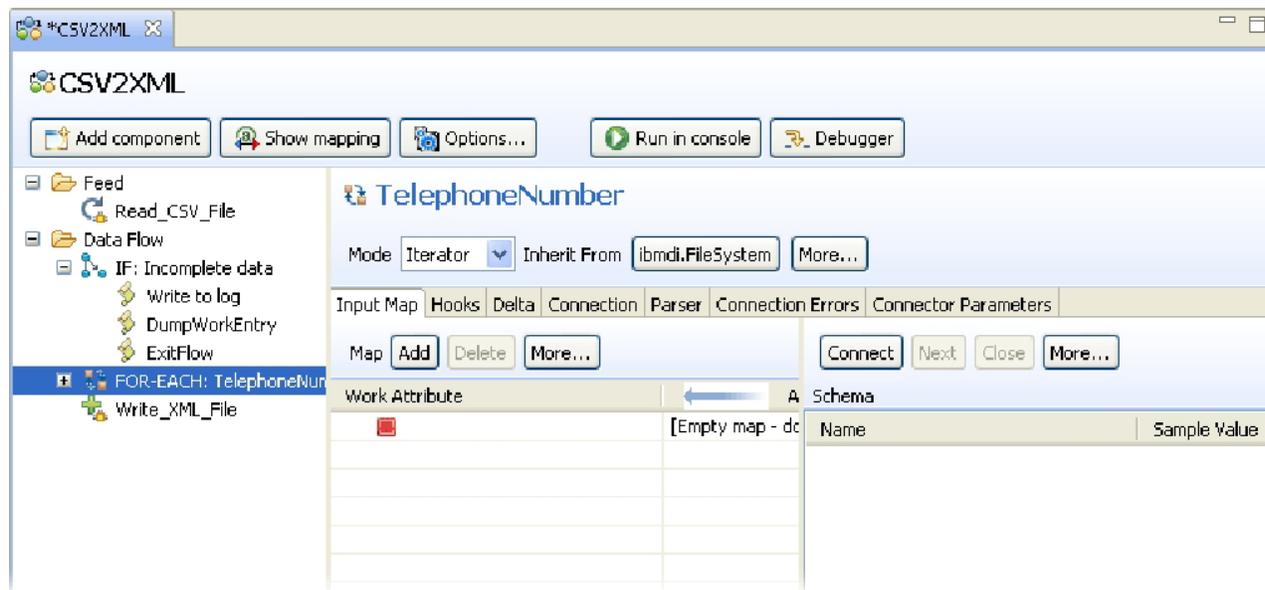


Figura 54. Configuração de ConnectorLoop

As principais diferenças são que o menu suspenso **Modo** conterá sempre somente as opções **Agente Iterativo** e **Consulta**. Além disso, há um botão **Mais...** que fornece opções para limitar as entradas em ciclo, assim como um parâmetro de menu suspenso **Inicializar** com as três seleções:

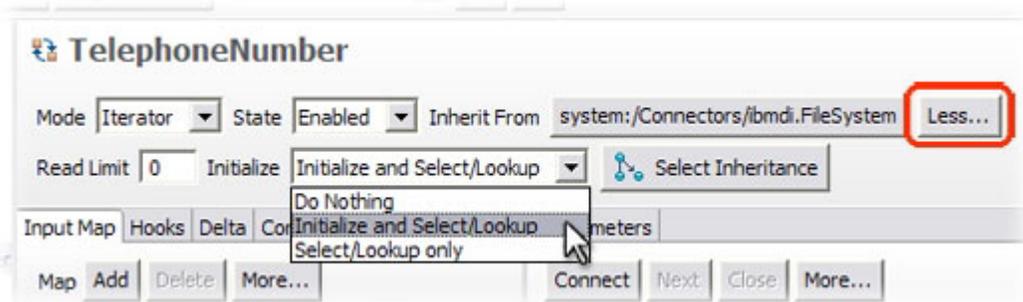


Figura 55. Configurações Avançadas de ConnectorLoop

- **Não Executar Nada**, o que significa que quando esse Loop for atingido durante o processamento do AL, seu Conector integrado não será inicializado de nenhuma maneira;
- **Inicializar e Selecionar/Consultar**, para fazer com que o Conector seja inicializado sempre que o Loop iniciar o ciclo. Use essa opção porque seu ConnectorLoop lerá um arquivo e você desejará que ele inicie sempre a partir do começo;
- **Somente Selecionar/Consultar**, que é útil quando seu ConnectorLoop está apontando um banco de dados, diretório ou alguma outra origem de dados acessível aleatoriamente. Não é necessário reinicializar a conexão toda vez neste caso. Tudo o que deve ser feito é emitir novamente a procura, que é *Selecionar* no caso do modo Agente Iterativo e uma operação *Consultar* para o modo Consulta.

Configure o LoopConnector (que é do tipo 'FileSystem' por padrão) para ler o arquivo PhoneNumber.xml e, em seguida, selecione o 'Analisador XML'. Agora, exiba a guia Mapa de Atributos para descobrir Atributos.

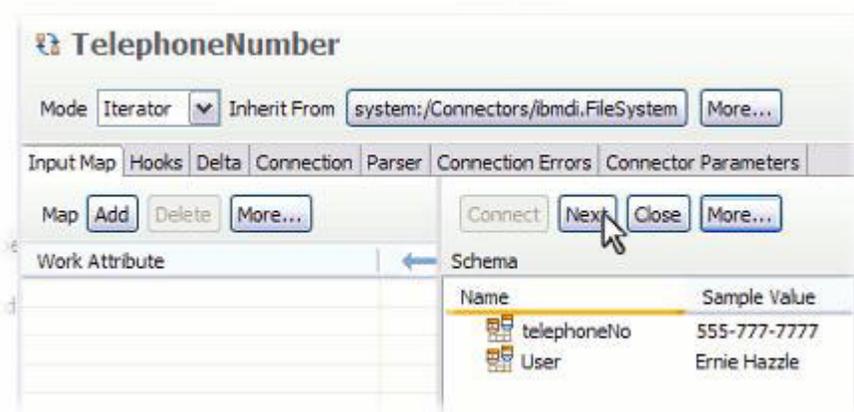


Figura 56. Atributos Hierárquicos

Você executará seu mapeamento no nível de Atributo, selecionando o 'User' e o 'telephoneNo' no Esquema de Entrada e arrastando-os para o Mapa de Entrada.

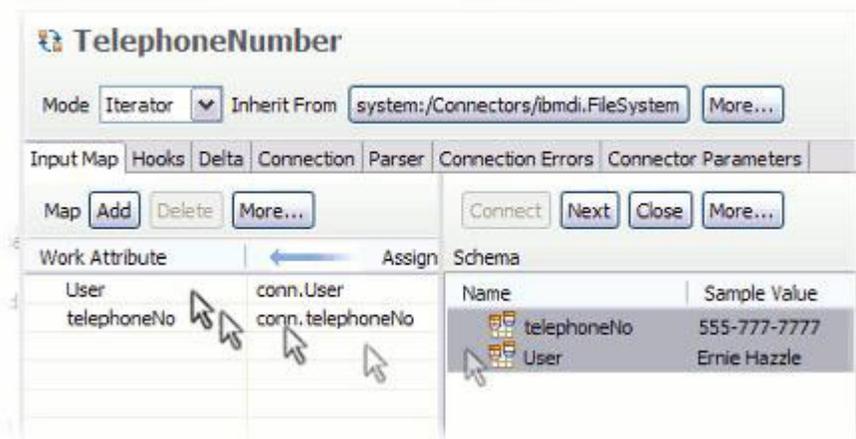


Figura 57. Arrastando do Esquema para o Mapa de Atributos

Como resultado, você terá uma regra de mapeamento para um Atributo nomeado "User" e uma para "telephoneNo".

Agora você pode fechar o editor de LoopConnector e, em seguida, incluir uma ramificação IF sob ele, clicando com o botão direito do mouse no ConnectorLoop e escolhendo **Incluir Componente...** Denomine essa ramificação IF como 'Nome correspondente localizado'. Agora, inclua uma condição simples que verifica se 'User' equivale a '\$FullName'²⁵.

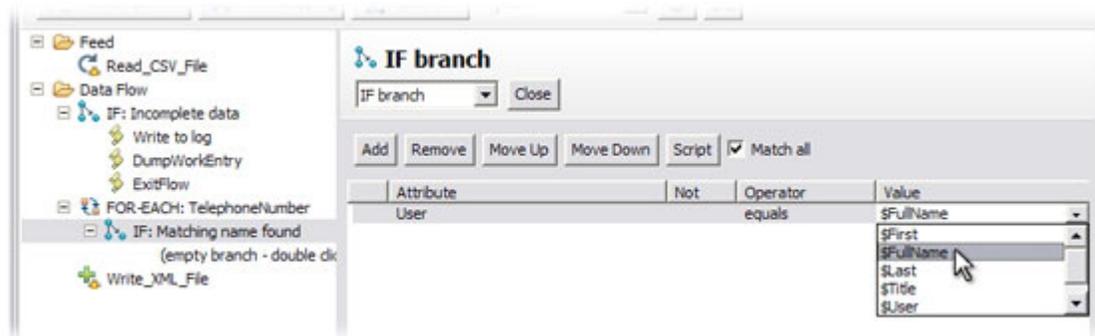


Figura 58. Editor de Condição para a Ramificação IF

Sempre que uma correspondência for localizada, você desejará que o ConnectorLoop saia com os valores corretos nos Atributos 'User' e 'telephoneNo'. Para fazer isso, inclua um componente de Script que você nomeia como 'Loop de Saída' e no qual grava o seguinte script:

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

Mas o que acontecerá se o ConnectorLoop atingir o fim do PhoneNumbers.xml sem localizar uma correspondência? Os Atributos 'User' e 'telephoneNo' contêm os valores lidos da última entrada no arquivo, portanto apenas verificar os Atributos vazios não ajudará. Você precisará descobrir alguma outra maneira de detectar uma correspondência com falha.

25. O cifrão é um caractere especial usado aqui para indicar que 'FullName' não é uma cadeia literal para corresponder, mas em vez disso, o valor de um Atributo localizado na Entrada de Trabalho.

A resposta é usar uma variável do script como um sinalizador para indicar que uma correspondência foi localizada. Faça isso inserindo um componente de Script que você nomeia como 'Usuário localizado' na ramificação IF, arrastando-o logo antes do SC 'Loop de Saída'. Esse componente de Script deverá conter o seguinte trecho de script:

```
foundUser = true;
```

Para indicar que o final do arquivo de entrada foi atingido sem localizar uma correspondência, simplesmente selecione seu ConnectorLoop e abra a guia Ganchos.

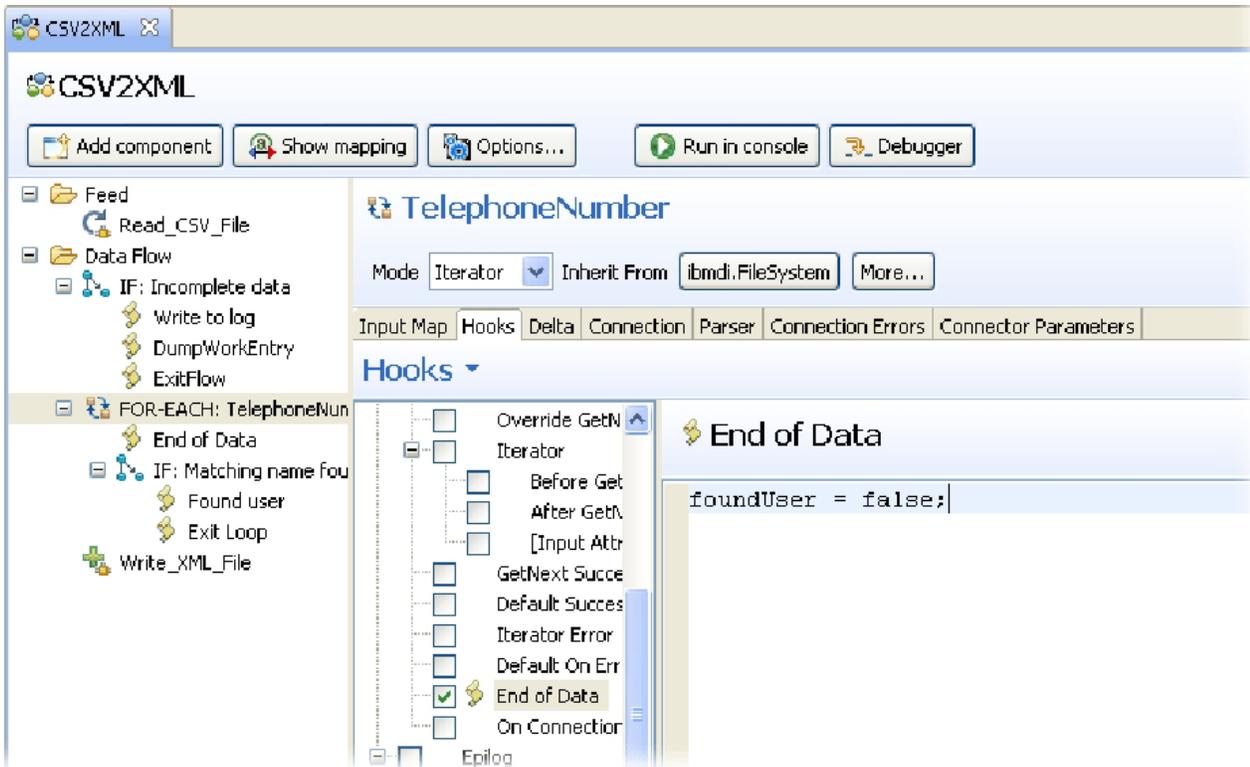


Figura 59. Script do Gancho Fim dos Dados

Selecione o Gancho denominado 'Fim dos Dados' e entre nesse script.

```
foundUser = false;
```

O Gancho 'Fim dos Dados' será atingido somente se o Conector tentar ler após a última entrada em sua origem conectada. Neste caso, nenhuma correspondência foi localizada.

Agora, seu AssemblyLine deverá ter estes componentes:

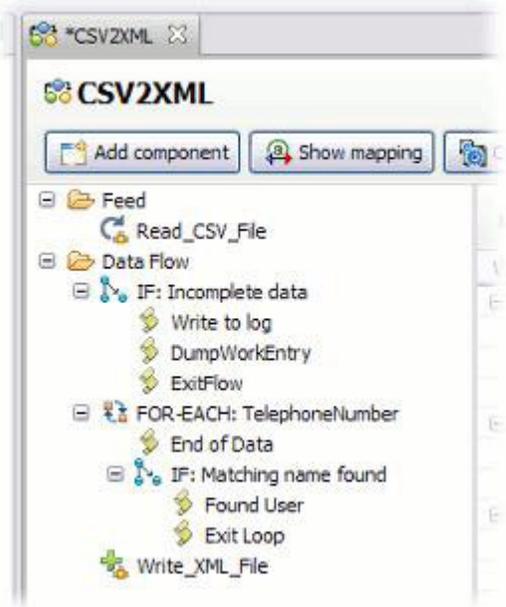


Figura 60. Lista de Componentes na seção Fluxo de Dados do AssemblyLine

Agora você deverá estar apto a averiguar se a procura obteve êxito, ou não, verificando a variável do script. Isso é importante porque sempre que nenhuma correspondência for localizada, você também deverá configurar um valor-padrão para o Atributo 'telephoneNo'; caso contrário, ele ainda terá o último valor lido pelo ConnectorLoop.

Portanto, inclua uma outra ramificação IF imediatamente após o ConnectorLoop e denomine-a como 'NÃO foundUser'. Clique no botão **Script** na área de detalhes da **Ramificação IF** e insira este script para verificar o valor de sua variável de script:

```
! foundUser
```

O ponto de exclamação nega o valor de foundUser, portanto, se ele tiver sido configurado como *false* no Gancho 'Fim dos Dados' de seu ConnectorLoop, essa Condição de ramificação será avaliada como *true*.

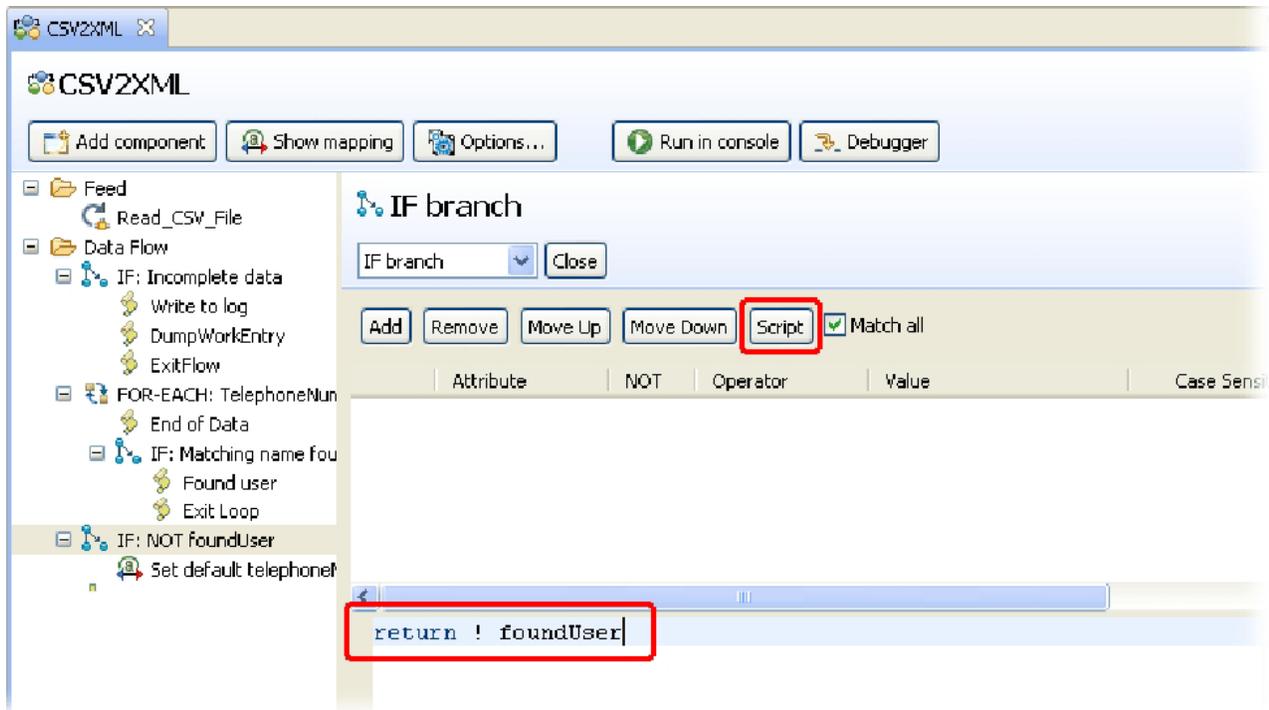


Figura 61. Script de uma Condição para a Ramificação IF

Insira um novo componente do tipo 'Mapa de Atributos' sob ele. Denomine esse componente de Mapa de Atributos como 'Configurar telephoneNo padrão' para tornar sua função clara no contexto de seu AssemblyLine. Agora, use o botão **Incluir Atributo** para criar um único Atributo nomeado 'telephoneNo' – o mesmo nome que aquele que está sendo retornado por seu ConnectorLoop. Clique duas vezes nesse Atributo para configurar o script de designação:

```
"N/A"
```

Isso significa que qualquer pessoa lida a partir de sua entrada CSV e não localizada em PhoneNumbers.xml obterá um valor de 'telephoneNo' igual a "N/A"²⁶.

Por último, inclua este novo Atributo 'telephoneNo' no Mapa de Saída de 'Write_XML_File', arrastando-o para lá e, em seguida, certificando-se de que a designação seja:

```
work.telephoneNo
```

Agora, seu AssemblyLine deverá ser semelhante a este.

26. Se você preferir que esses usuários não tenham o Atributo 'telephoneNo', simplesmente use uma designação que não retorne valor. Isso é feito retornando o valor especial nulo:

```
nulo
```

Isso fará com que o Comportamento Nulo padrão remova o Atributo da Entrada de Trabalho. Como resultado, ele não atingirá o Mapa de Saída de seu Conector 'Write_XML_File' e, portanto, não aparecerá no documento XML resultante.

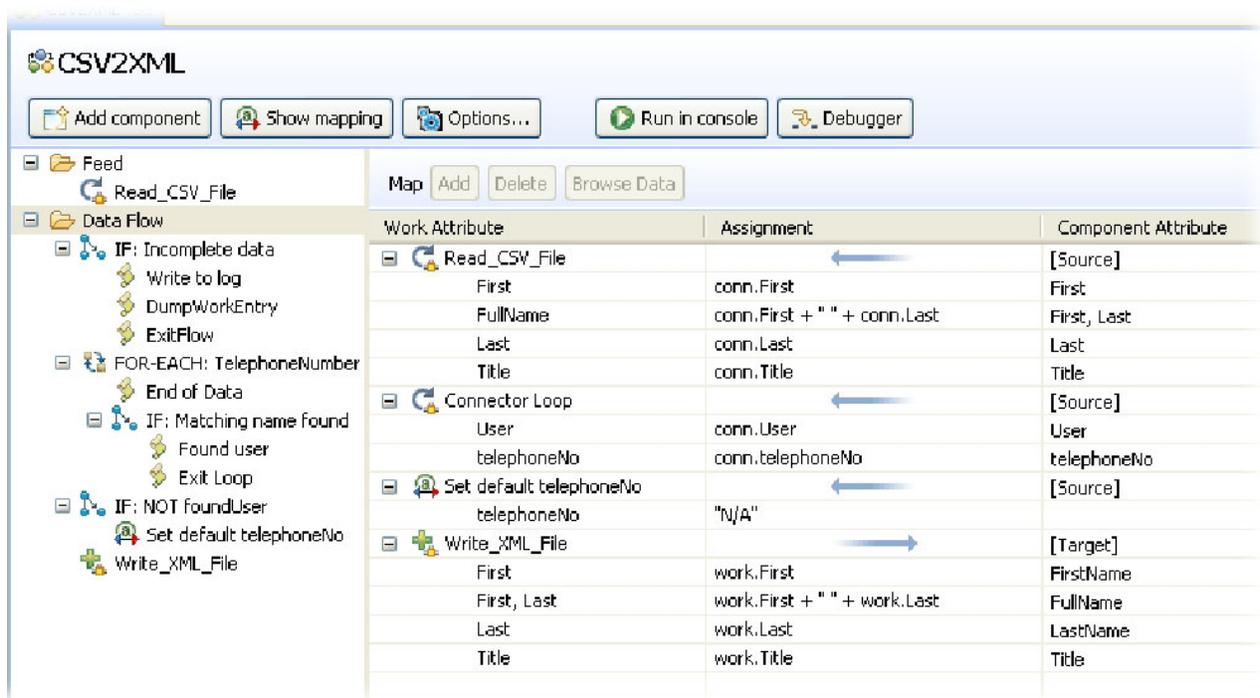


Figura 62. AssemblyLine Completado com Loop FOR-EACH

Agora, execute seu AL novamente e examine a saída do log. Sua ramificação 'NÃO foundUser' deverá ter sido *true* duas vezes e *false* para as outras quatro entradas.

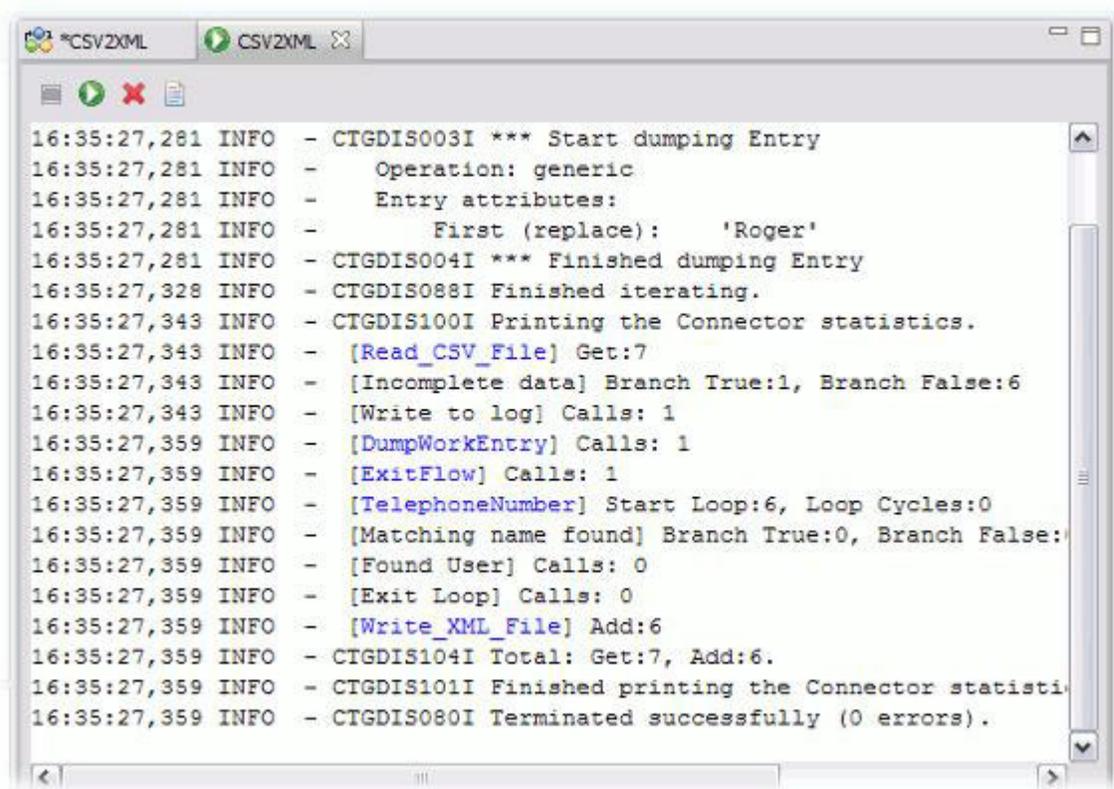


Figura 63. Saída do Log com Estatísticas da Ramificação IF

Observe que alguns nomes de componentes estão realçados (em azul) nas estatísticas do AL da saída do log. Se você usar Ctrl-clique em algum com o botão esquerdo do mouse, ele abrirá o componente selecionado acima no editor de AssemblyLine.

Como resultado, sua saída XML deverá ser semelhante a esta:

```
-<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. Saída XML com Atributo 'telephoneNo'

Até aqui, tudo bem. Agora é o momento de tentar usar o modo *Consulta* para executar a junção.

Usando o Modo de Consulta

Em vez de modificar e, potencialmente, desordenar seu AssemblyLine em execução, você fará uma cópia dele e, então, alterará a cópia. Faça isso clicando com o botão direito do mouse em 'CSV2XML.assemblyline' e selecionando **Copiar**.

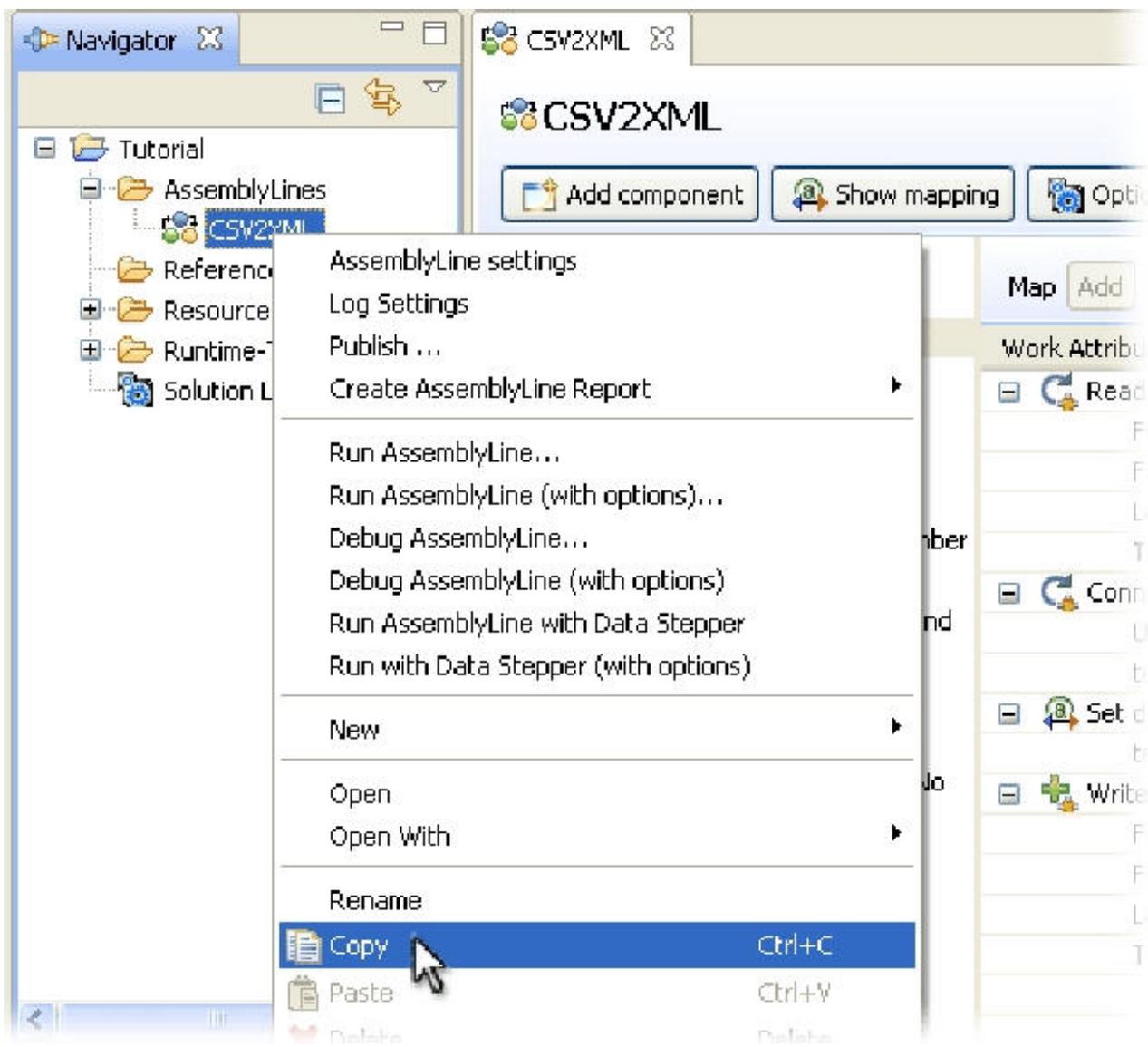


Figura 65. Função Copiar do AssemblyLine

Agora, clique com o botão direito do mouse na pasta 'AssemblyLines' no Navegador e selecione a opção **Colar**. Denomine esse novo AssemblyLine como 'CSV2XML_LookupMode' e, em seguida, clique duas vezes nele para abrir o editor de AL.

Agora, você pode remover o ConnectorLoop juntamente com todos os componentes sob ele. Simplesmente selecione-o e pressione a tecla **Excluir**. Em seu lugar, você arrastará um Conector JDBC que será copiado de um outro AssemblyLine.

Mas, primeiramente, você deve construir a tabela de banco de dados a partir da qual esse Conector fará a leitura; ou preferivelmente, deve executar um AssemblyLine pré-construído para construí-lo para você. Para fazer isso, use um navegador de arquivos para navegar para a pasta 'Tutoriais' e localize o arquivo chamado CreatePhoneDB.assemblyline. Arraste-o para a janela CE sobre a pasta 'AssemblyLines' no painel Navegador.

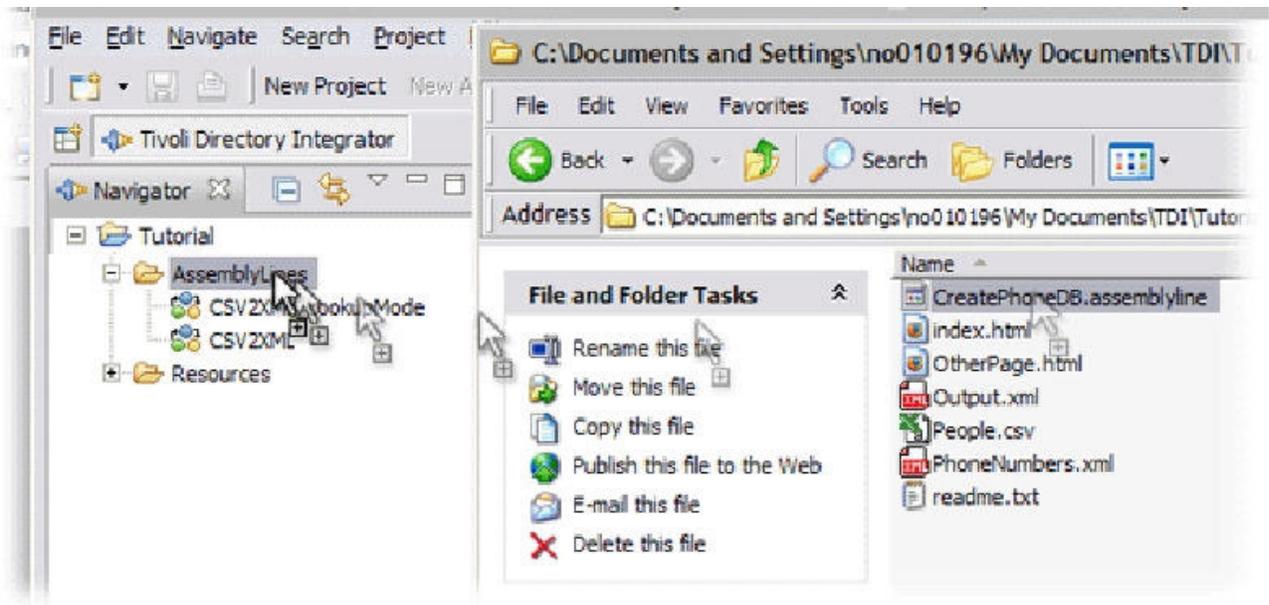


Figura 66. Copiando um AssemblyLine para seu Projeto

O AssemblyLine será importado para seu projeto. Agora, clique com o botão direito do mouse nele e escolha **Executar AssemblyLine...**

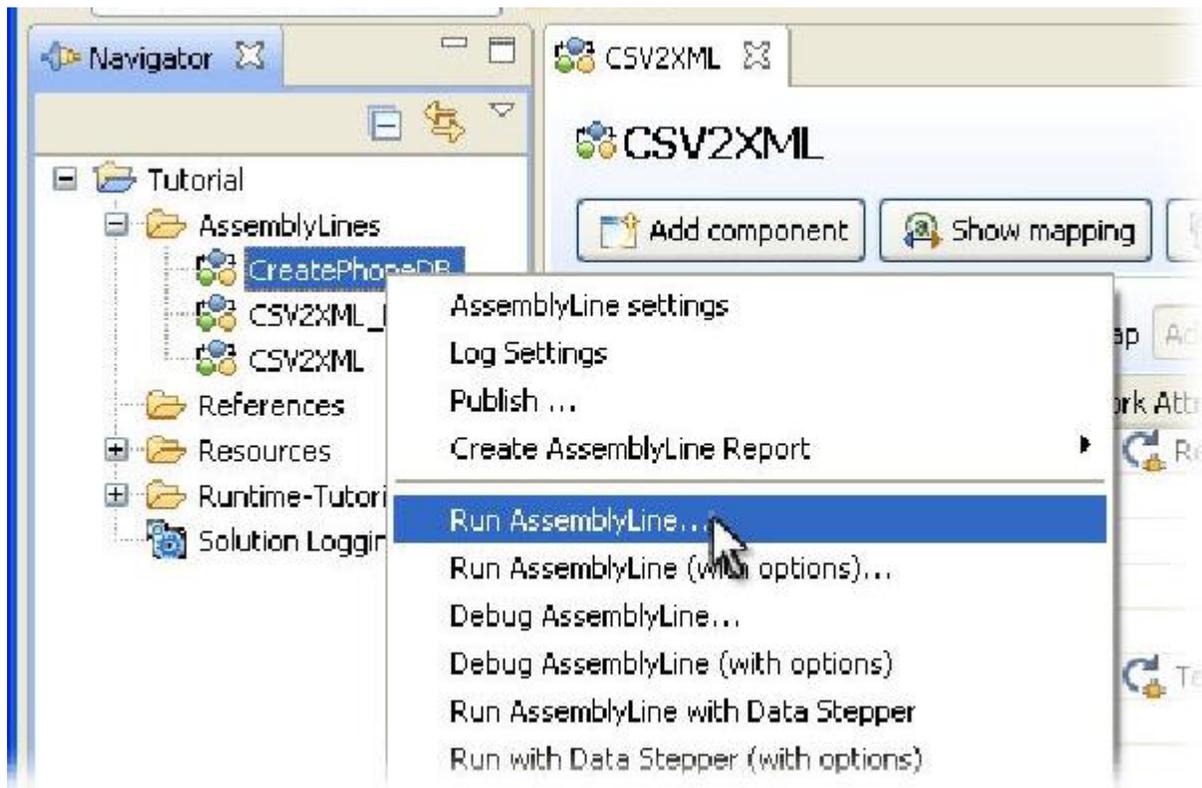
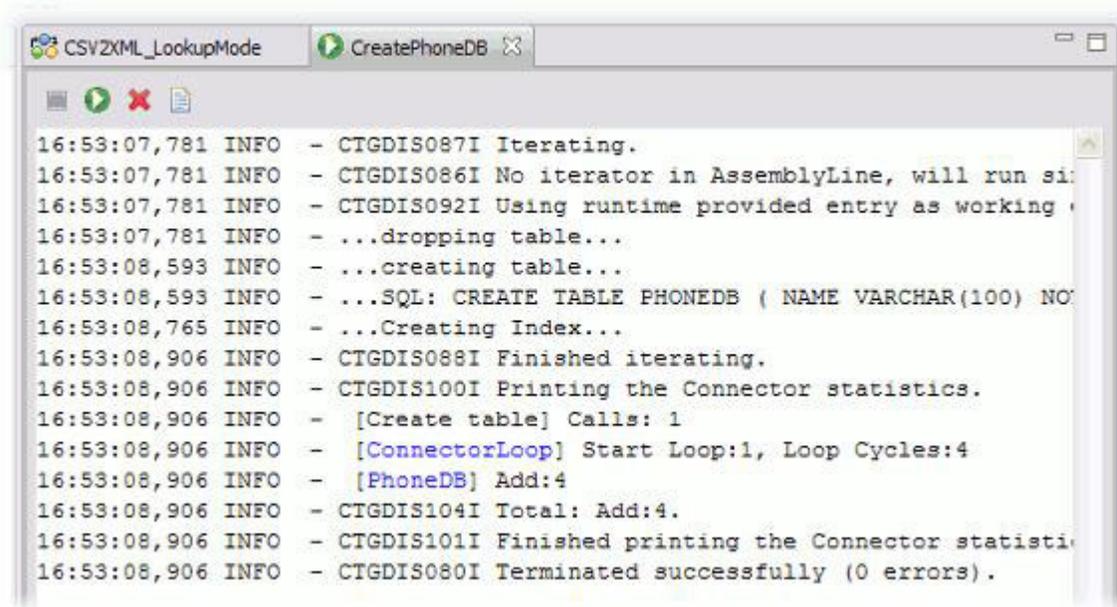


Figura 67. Executar o AL CreatePhoneDB

Esse AssemblyLine criará primeiro um Derby²⁷ sob seu diretório de solução chamado 'TutorialDB' e, depois, configurará uma tabela 'PhoneDB'. Ele efetuará, então, um loop no PhoneNumbers.xml e carregará essas informações na nova tabela²⁸.

Se tudo estiver satisfatório, a saída do log deverá ser semelhante a esta:



```
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. Saída do Log a partir do AssemblyLine 'CreatePhoneDB'

O AssemblyLine 'CreatePhoneDB' possui um Conector JDBC que já está configurado e pronto para uso. Você copiará esse componente para sua biblioteca de Recursos do Projeto (especificamente para a pasta "Conectores") e, em seguida, o reutilizará em seu AL.

Abra o AssemblyLine 'CreatePhoneDB', pegue o Conector chamado 'PhoneDB' e arraste-o para a pasta 'Conectores' localizada sob 'Recursos' na árvore do Navegador.

27. Apache Derby é um banco de dados relacional de software livre, empacotado com o IBM Security Directory Integrator.

28. Embora o AL não seja abrangido neste guia, este é um bom exemplo das técnicas de script avançado usadas para explorar a funcionalidade específica da origem de dados localizada na maioria dos Conector Interfaces. Sinta-se à vontade para examinar e reproduzir isso com este AssemblyLine.

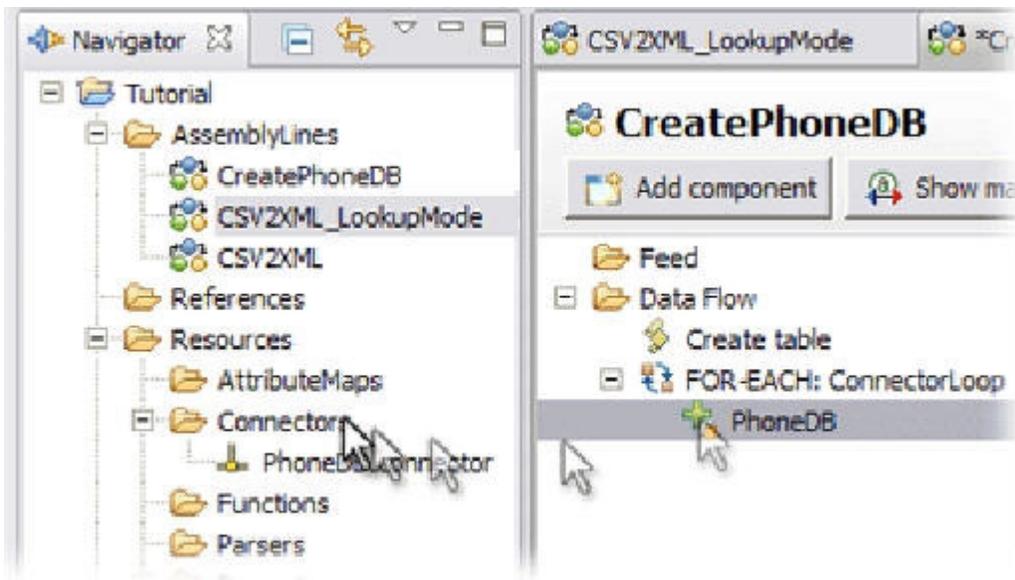


Figura 69. Arrastar um Conector para Recursos

Feche o 'CreatePhoneDB' para que seu próprio AssemblyLine fique visível novamente. Em seguida, arraste o novo recurso 'PhoneDB.connector' para o ponto ocupado anteriormente pelo ConnectorLoop.

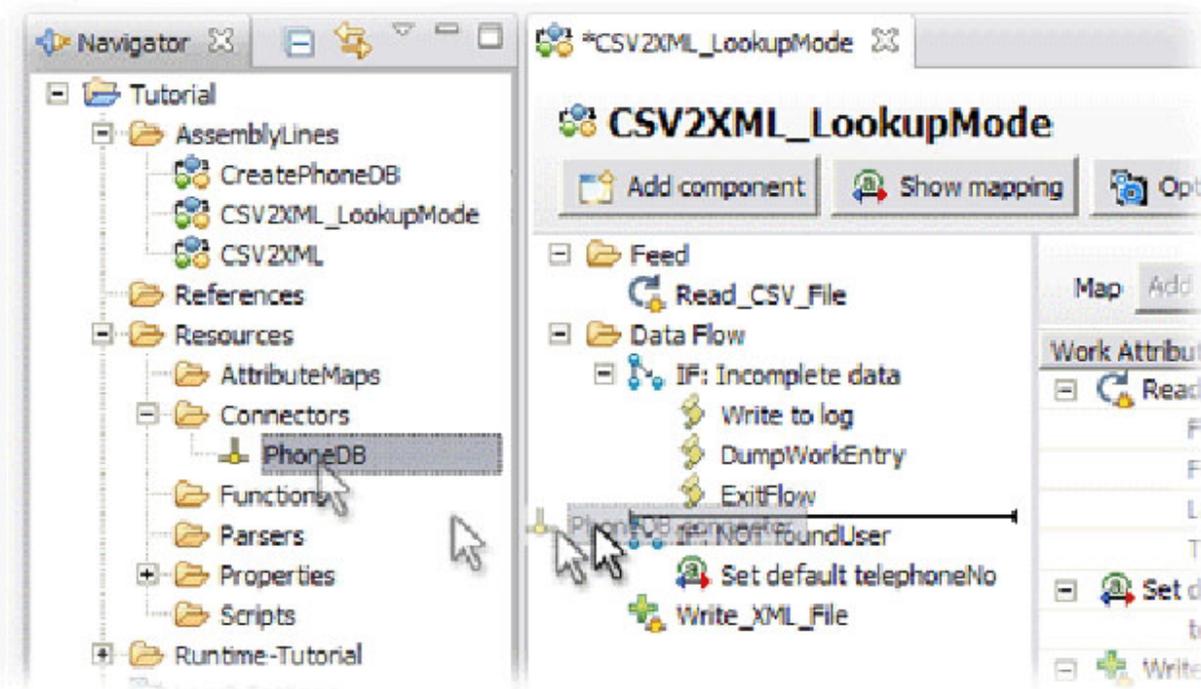


Figura 70. Arrastar o Novo Recurso para seu AssemblyLine

Herança

Você observou que este Conector aparece em azul em seu AssemblyLine? Isto ocorre porque agora ele está herdando de sua biblioteca de recursos. Isto significa

que ele recuperará dinamicamente as definições de configuração no tempo de execução do Conector arrastado. Esse recurso de herança facilita a reutilização de recursos, como componentes configurados e lógica de script entre múltiplos AssemblyLines.

É possível alterar o ancestral de um componente com o botão **Herdar de** na parte superior de seu painel do editor. As guias de componentes, como Configuração, Delta, Mapas de Entrada/Saída e Ganchos, também fornecem uma opção de herança, permitindo configurar um ancestral diferente daquele do próprio componente.

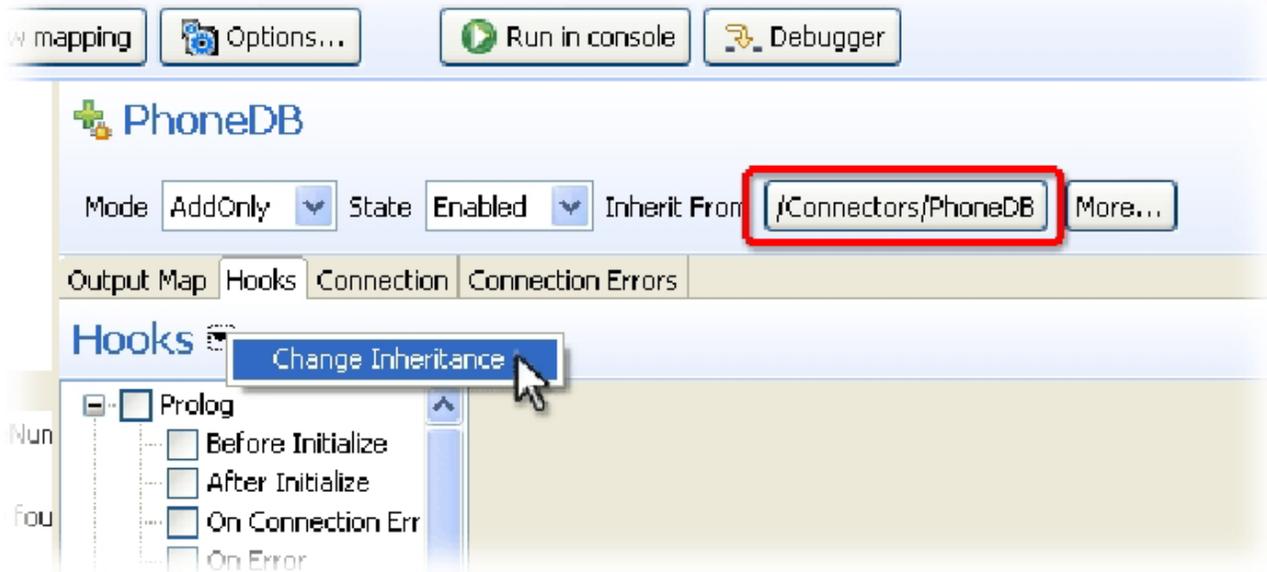


Figura 71. Configurando a Herança para a Guia Ganchos

Os valores herdados são exibidos no tipo azul e, se você alterá-lo, a herança será dividida. A herança é restaurada usando a opção **Reverter para Valor Herdado** nos menus de contexto das regras do Mapa de Atributos e dos Ganchos.

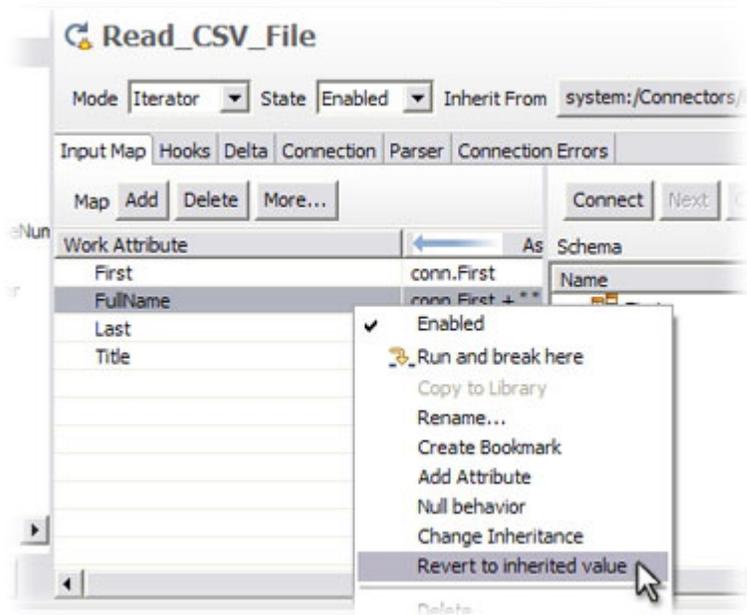


Figura 72. Restaurando a Herança para uma Regra de Mapeamento

Retornando ao exercício novamente, em primeiro lugar, altere a configuração **Modo** do novo Conector "PhoneDB" de **AddOnly** para **Consulta**.

Em segundo lugar, como o Mapa de Atributos era originalmente um *Mapa de Saída* associado ao modo anterior, você precisará descobrir o esquema de entrada pressionando os botões **Conectar** e **Próximo** acima do Esquema do Conector. A terceira e última etapa é arrastar o Atributo 'PHONE' do Esquema sobre o Mapa de Entrada, fornecendo um mapa simples para esse valor.

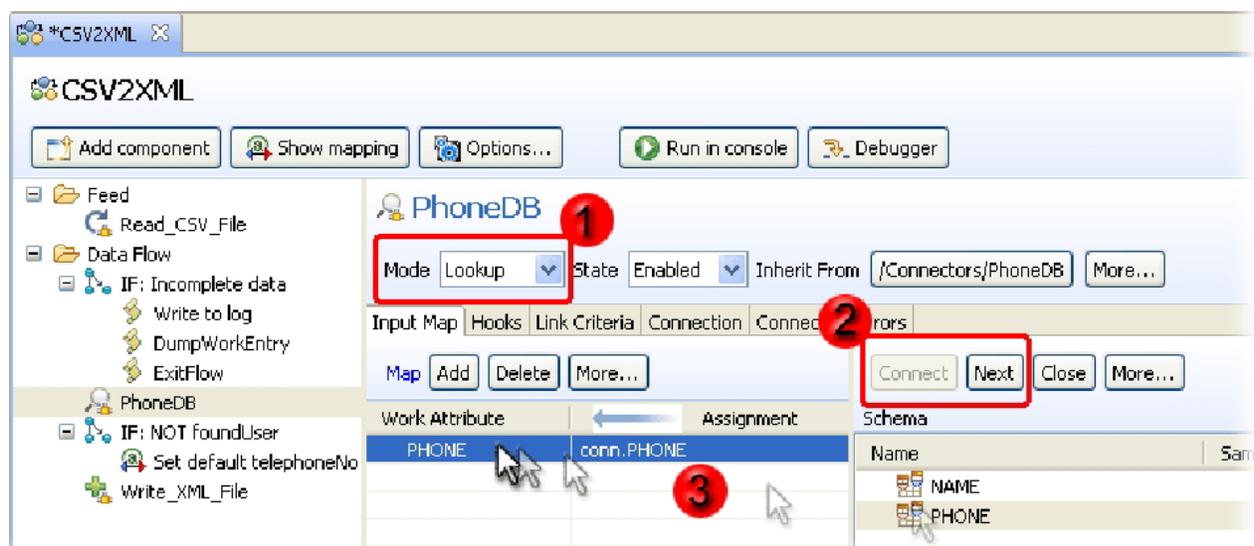


Figura 73. Alterando o Modo, Descobrimo e Mapeando Atributos

Uma vez que você possui a regra do Mapa de Entrada no local, é importante renomeá-la de 'PHONE' para 'telephoneNo' para que se ajuste à designação do Mapa de Saída de seu Conector 'Write_XML_File'.

E caso esteja curioso por saber, você não precisa mapear 'NAME' pois já possui o nome do usuário. Em vez disso, esse campo será usado ao definir a regra de procura.

Consultar Regras de Procura = Critérios de Link

Quando você selecionou o modo de *Consulta*, uma nova guia rotulada 'Critérios de Link' apareceu no editor de Conector. Os Critérios de Link são para definir as regras de correspondência para a procura.

Você pode definir estes como Critérios de Link simples, usando os menus suspensos, incluindo novos Critérios de Link, conforme necessário, e configurando a caixa de opção **Corresponder Qualquer Um**, tal como faria para Condições. Alternativamente, você pode selecionar a caixa de opção **Construir critérios com script customizado** e, em vez disso, gravar um trecho de script que computa a regra de procura real, como este exemplo de um filtro de procura LDAP:

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

Observe que isto ligará sua solução mais estreitamente com a origem de dados sendo procurada, pois você precisa gravar a sintaxe real esperada pelo sistema conectado. Em nosso caso, significaria criar uma cláusula WHERE (sem a própria palavra-chave 'WHERE').

Em contraste, Critérios de Link simples são convertidos em sintaxe de procura nativa pelo Conector, portanto, você pode alternar o Connector Interface sem precisar refazer seus Critérios de Link.

Os Critérios de Link Simples são semelhantes a Condições. O primeiro menu suspenso é preenchido com o esquema descoberto e o segundo mostra quais Atributos de Entrada de Trabalho estão disponíveis neste ponto em seu AL. Novamente, tal como ocorre com as Condições, o cifrão é usado aqui para indicar que o valor do Atributo nomeado deve ser substituído no tempo de execução para criar o filtro de procura.



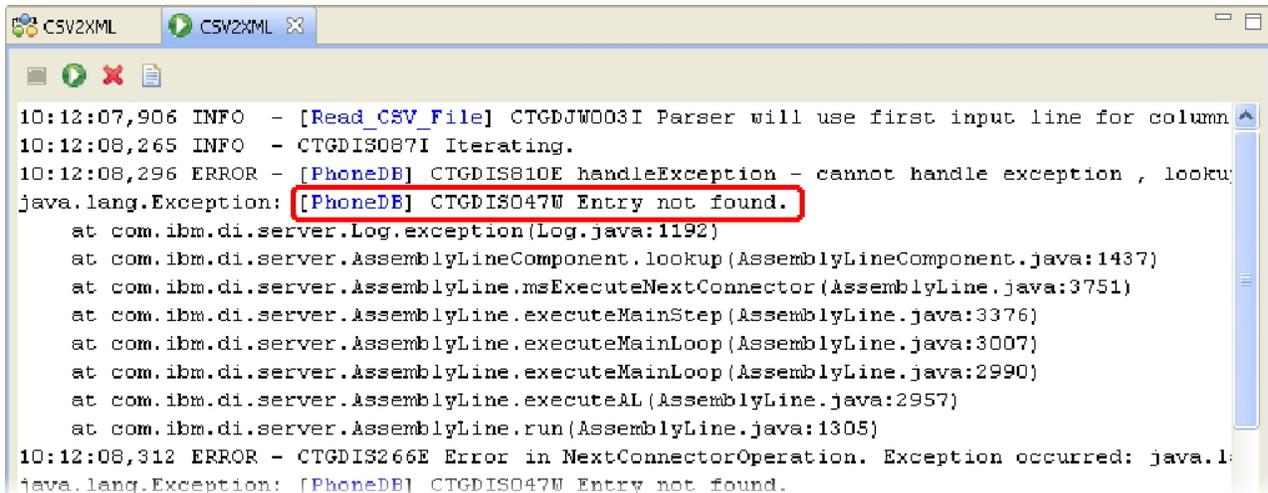
Figura 74. Um Critério de Link Simples

Lembre-se de salvar seu trabalho, por exemplo, pressionando CTRL-S²⁹. Você desejará fazer isso regularmente para que não perca nenhum trabalho.

É o momento de Executar seu AssemblyLine novamente.

Decifrando Erros de Execução

Não se preocupe. Você deve ter recebido um erro, indicado pelo dump de pilha que aparece na saída do log. Role para a parte superior do primeiro dump de pilha. Aqui, você verá informações sobre *onde* o problema ocorreu e também *o que* o causou.



```
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] CTGDIS10E handleException - cannot handle exception , looku
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. Mensagem de Erro na Saída do Log

O nome do componente está entre colchetes ('PhoneDB'), seguido pela descrição do erro de que uma Entrada não foi localizada – ou seja, que a Consulta falhou.

Quando um Conector está configurado no modo de *Consulta*, o sistema espera localizar somente um registro correspondente quando a procura é executada. Se nenhum for localizado – ou se múltiplos registros corresponderem aos Critérios de Link – você irá parar em *Ganchos* especiais, que devem ser pelo menos ativados para evitar que o AL pare. Esse comportamento fica claramente visível nos diagramas de Fluxo de Dados que fazem parte do *Referência*. Este é um fragmento da página detalhando o modo de *Consulta*:

29. Se você excluiu os AssemblyLines ou recursos e desejar desfazer isso, clique com o botão direito do mouse em seu projeto no painel do Navegador e selecione **Restaurar a partir do histórico local...**, que apresentará uma lista de versões de ativos dos quais é possível restaurar. Naturalmente, você precisará salvar algo primeiro antes que isso seja mostrado em seu Histórico Local.

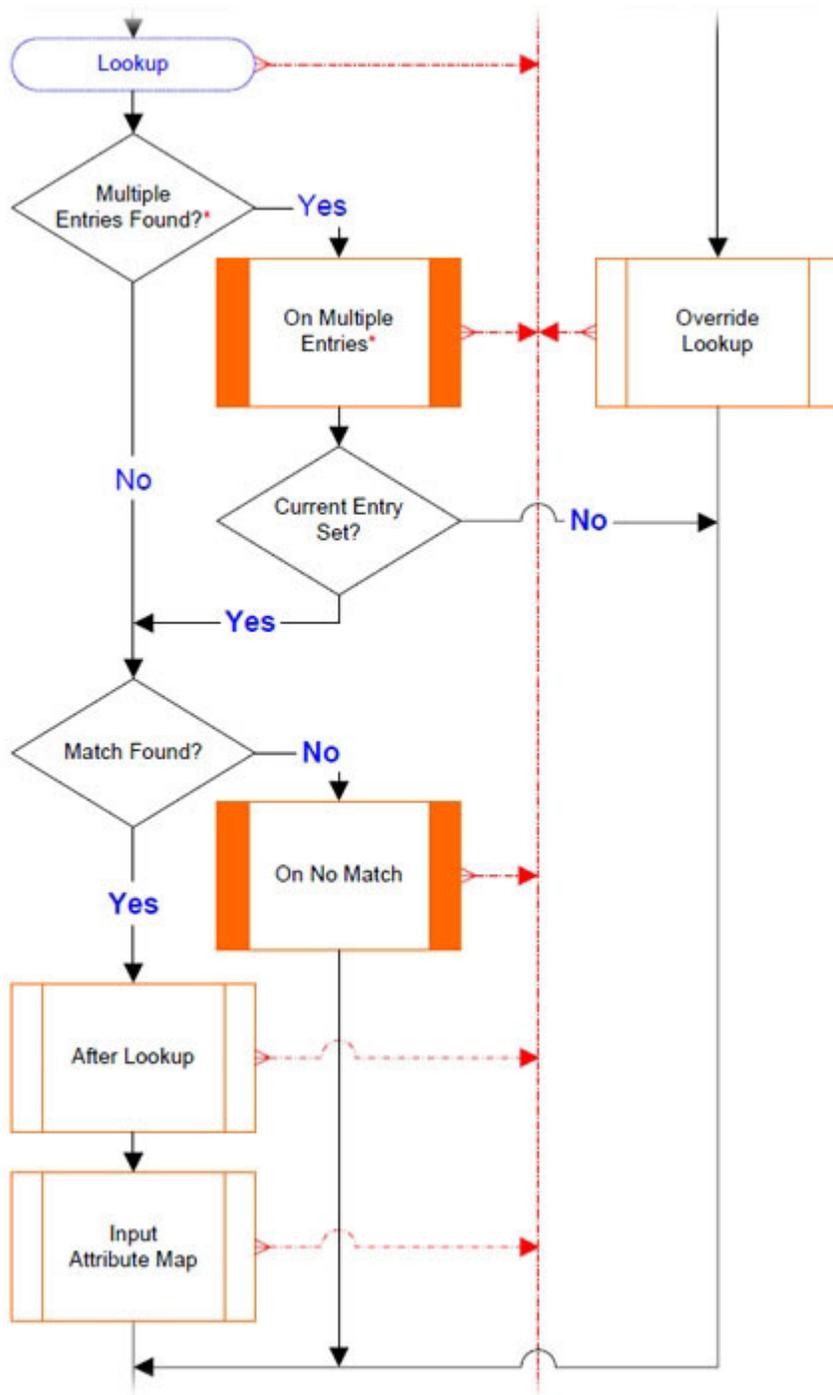


Figura 76. Fluxograma Parcial para o Modo de Consulta

Você pode aproveitar a vantagem desse comportamento para configurar sua variável de sinalizador `foundUser`. Clique com o botão direito do mouse no Conector 'PhoneDB' e escolha **Ganchos...** para abrir o editor de Ganchos. Selecione o Gancho 'Em Nenhuma Correspondência' e insira o código de script para configurar `foundUser` como *false*.

```
foundUser = false;
```

Agora escolha 'Após Consulta' e insira esse script de Gancho complementar³⁰:
foundUser = true;

Agora, seu AssemblyLine deverá ser semelhante àquele na captura de tela:

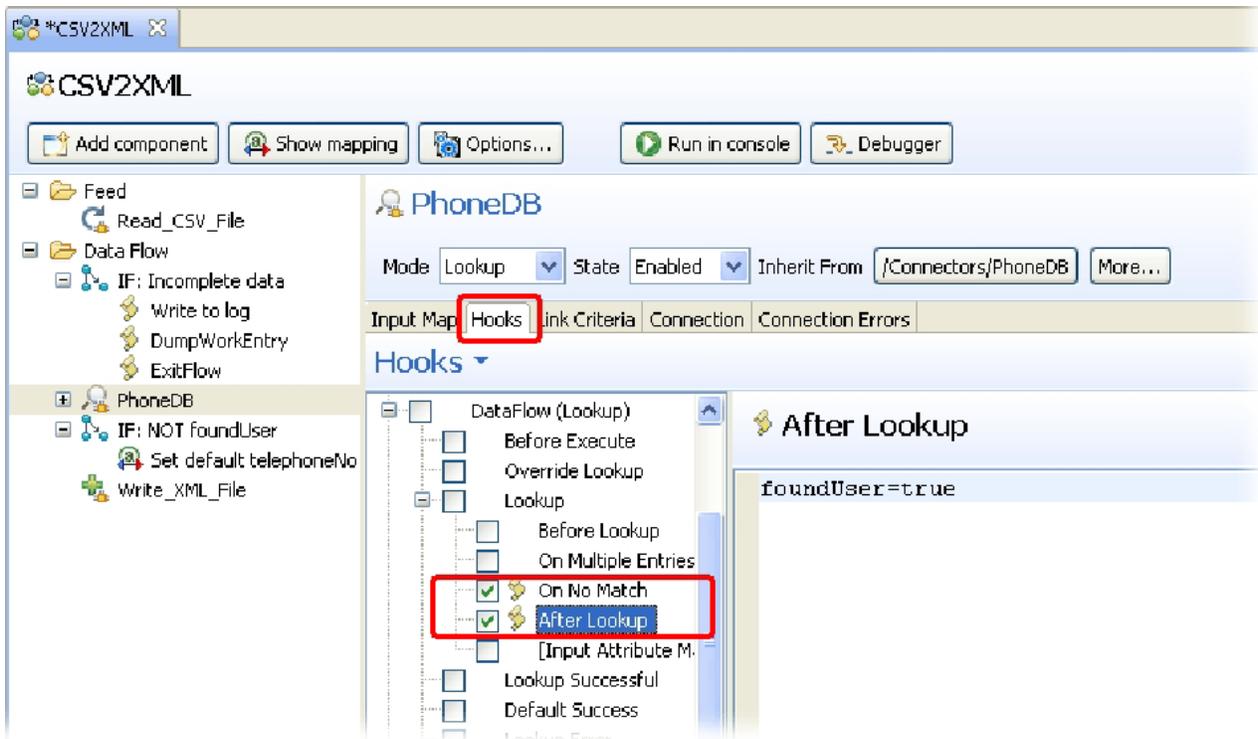


Figura 77. Primeiro Exercício do Tutorial Concluído

É o momento de **Executar** esse AssemblyLine novamente e, desta vez, ele deve ser completado sem erros. O arquivo `Output.xml` deve ser idêntico aos resultados obtidos de seu AL baseado em Loop.

Parabéns! Você acabou de concluir seu primeiro exercício do tutorial do IBM Security Directory Integrator. Agora é o momento de discutir como os AssemblyLines podem ser acionados por eventos em tempo real.

30. Como você pode ver no Fluxograma, o Gancho 'Após Consulta' será executado somente se a procura resultar em uma única correspondência.

Capítulo 3. Integração Acionada por Eventos

Até aqui, você executou seus AssemblyLines como processos em lote, iniciando-os manualmente toda vez que deseja que os dados fluam. Também é possível executar AssemblyLines a partir da linha de comandos, chamando o Servidor IBM Security Directory Integrator, como este³¹:

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

Desta maneira, é uma tarefa simples usar as ferramentas de planejamento (por exemplo, *crontab*) para planejar suas operações ou para ativá-las facilmente a partir de aplicativos externos.

O IBM Security Directory Integrator fornece vários recursos para tornar os AssemblyLines cientes de eventos, permitindo que suas soluções manipulem e respondam a uma grande variedade de acionadores em tempo real.

Estes são exemplos desses acionadores:

- pedidos de protocolo provenientes através de uma porta IP, como chamadas REST, SNMP e serviços da Web;
- novas mensagens que aparecem em uma fila;
- e-mails que chegam em uma Caixa de Entrada;
- mudanças nos dados, por exemplo, em arquivos, bancos de dados, diretórios e bancos de dados do Notes;
- operações do AssemblyLine baseadas no planejamento ou no cronômetro.

Essa não é uma lista completa e você encontrará inspiração e orientação em outras publicações do IBM Security Directory Integrator, nos Web sites de comunidades e nos grupos de notícias.

A manipulação desses eventos em sua solução pode ser feita de várias maneiras:

Conectores no Modo de Agente Iterativo

Alguns Conectores permitem que você configure parâmetros de tempo limite para o Modo de Agente Iterativo. Um exemplo é o Conector FileSystem, que pode ser configurado para ler um arquivo até o final e, então, aguardar que novas informações apareçam - chamado de 'leitura até o final'.

Outros Conectores, como aqueles para detecção de Mudança do RDBMS e Log de Mudanças do LDAP, funcionam de um modo semelhante. Esses Conectores permitem que você construa AssemblyLines, que executam continuamente e aguardam que novas alterações apareçam no sistema conectado.

Há também um Conector Cronômetro (Timer) que executa no Modo Agente Iterativo (Iterator) e pode ser configurado para orientar o AssemblyLine em intervalos com tempo determinado de acordo com um parâmetro de planejamento. Você o testará em breve.

31. O Servidor IBM Security Directory Integrator fornece uma mensagem de uso quando chamado sem nenhum argumento de linha de comandos: `ibmdisrv`

O IBM Security Directory Integrator inclui um console de Administração da Web como parte da instalação padrão. Esse aplicativo baseado no navegador, chamado de Administration and Monitoring Console (AMC), permite monitorar o funcionamento de seus AssemblyLines, carregar a quente as Configurações para Servidores em execução, iniciar e parar ALs e configurar o comportamento de falha/resposta para manter suas soluções de integração altamente disponíveis. Também é possível usá-lo para configurar planejamentos para quando os AssemblyLines devem ser executados. Entretanto, a ferramenta Web Admin está além do escopo deste Guia.

Nota: O recurso AMC foi descontinuado e será removido em uma versão futura do IBM Security Directory Integrator.

Conectores no Modo Servidor

Alguns Conectores especializados, como o Conector do Servidor HTTP e o Conector do Servidor LDAP, permitem que você construa soluções que processam pedidos recebidos de clientes externos, executam ações solicitadas e respondem com respostas apropriadas. Você usará o Conector do Servidor HTTP no próximo exercício.

Notificações e Propriedades

O IBM Security Directory Integrator possui componentes que podem inscrever-se para eventos de notificação do IBM Security Directory Integrator, assim como há componentes (e chamadas de script) para enviar esses eventos – mesmo entre Servidores IBM Security Directory Integrator distintos sendo executados em plataformas diferentes.

Este guia guiará você por uma examinação minuciosa no planejamento do AL e no modo de Servidor do Conector.

Planejando AssemblyLines

Usando a interface com o usuário no Editor de Configuração, é possível criar um Planejador para executar o AssemblyLine em horários especificados. Por exemplo, é possível criar um Planejador para ser executado todos os dias às 3h05, todos os sábados às 7h ou até mesmo em horários mais complexos.

Para criar um Planejador, no Editor de Configuração, clique em **Arquivo > Novo > Planejador**. Como alternativa, clique com o botão direito do mouse no AssemblyLine e selecione **Criar Planejamento**.

Quando o Servidor IBM Security Directory Integrator carrega um arquivo de configuração, os Planejadores ativados no arquivo de configuração são iniciados automaticamente. O comando para carregar um arquivo de configuração é `ibmdisrv -c myconfig.xml -d`, em que `myconfig.xml` é o arquivo de configuração exportado. Parar a Instância de Configuração para todos os Planejadores associados.

Figura 78. Planejador do IBM Security Directory Integrator

Os detalhes do Planejador do IBM Security Directory Integrator são detalhados na seção *Planejador do IBM Security Directory Integrator* do *Referência*.

AssemblyLines de Pedido de Serviço

Neste último exercício, você construirá um AssemblyLine de servidor da Web para fornecer uma interface com o usuário bem simples para ativar seu AL 'CSV2XML_LookupMode'; ou seja, um serviço HTTP para iniciar transferências de dados.

Inicie criando um novo AL e denomine-o como 'TINA_WebServer'³². Em seguida, insira um novo componente, escolhendo o 'Conector de Servidor HTTP' e pressionando **Concluir** para terminar o assistente. Agora, abra sua guia **Mapa de**

32. TINA aqui significa "TINA Não é Apache" :-)

Entrada.

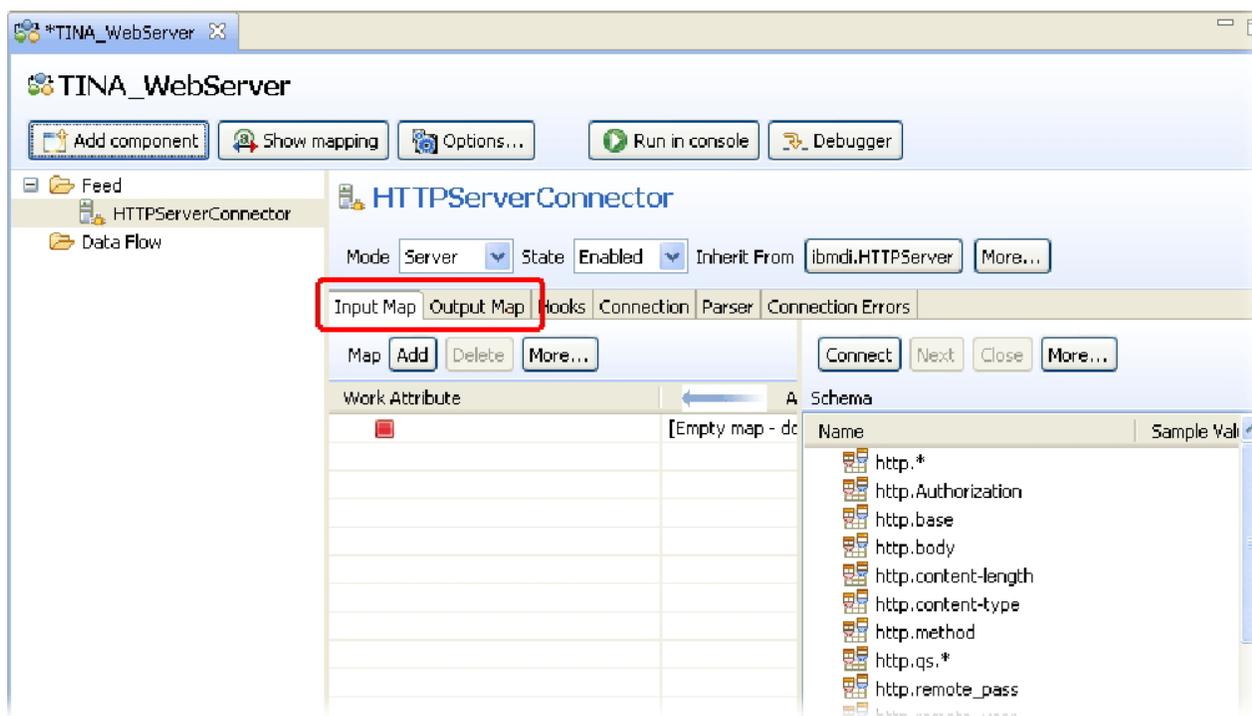


Figura 79. Painel Mapa de Atributos do Conector de Servidor HTTP

Os Conectores de Servidor são complementares aos Componentes de Função. Enquanto os FCs fazem pedidos de serviço, um Conector de Servidor fornece e potencializa um serviço. Como resultado, o Mapa de Entrada para um Conector de Servidor é usado para receber Atributos provenientes do cliente que está fazendo um pedido, enquanto o Mapa de Saída fornece um meio de resposta. Você também pode ver na parte à direita da tela Mapa de Atributos que o Esquema de Entrada já está no local. O mesmo é verdadeiro para o Esquema de Saída. Os Conectores de Servidor fornecem essas informações para ajudá-lo a executar seu mapeamento. Entretanto, observe que alguns desses Atributos de Esquema contêm caracteres curinga, como 'http.qs.*'. Estas são as informações de tempo de design que indicam a você para esperar qualquer número de Atributos recebidos cujos nomes iniciam com 'http.qs'.³³

Por último, o menu suspenso **Modo** para todos os Conectores de Servidor oferecem os modos Servidor e Agente Iterativo. Há um modo adicional (Resposta) não mostrado aqui, promovendo o total para três. O Conector de Servidor alterna entre esses modos em vários estágios de sua operação:

1. Um Conector de Servidor primeiro inicia no modo *Servidor*, conectando-se a algum recurso como uma porta IP e aguardando conexões do cliente recebidas;
2. Assim que uma conexão é feita, o Conector alterna para o modo *Agente Iterativo* para recuperar dados do cliente com base no Mapa de Entrada e passando isso para os componentes de Fluxo de Dados;

33. Esse conjunto específico de Atributos (http.qs.*) transportará quaisquer parâmetros de cadeia de consulta passados na chamada HTTP pelo cliente.

3. Por último, quando os componentes de *Fluxo de Dados* concluem a execução, o Conector entra no modo de Resposta, usando o Mapa de Saída para formar uma resposta de volta para o cliente.

Você não precisa se preocupar com isso, uma vez que ele é tratado automaticamente. Entretanto, se você executar algum script de Gancho, observará que há três conjuntos de Ganchos: *Servidor*, *Agente Iterativo* e *Resposta*.

Continue com o exercício do tutorial incluindo um item de Atributo do Mapa de Entrada.

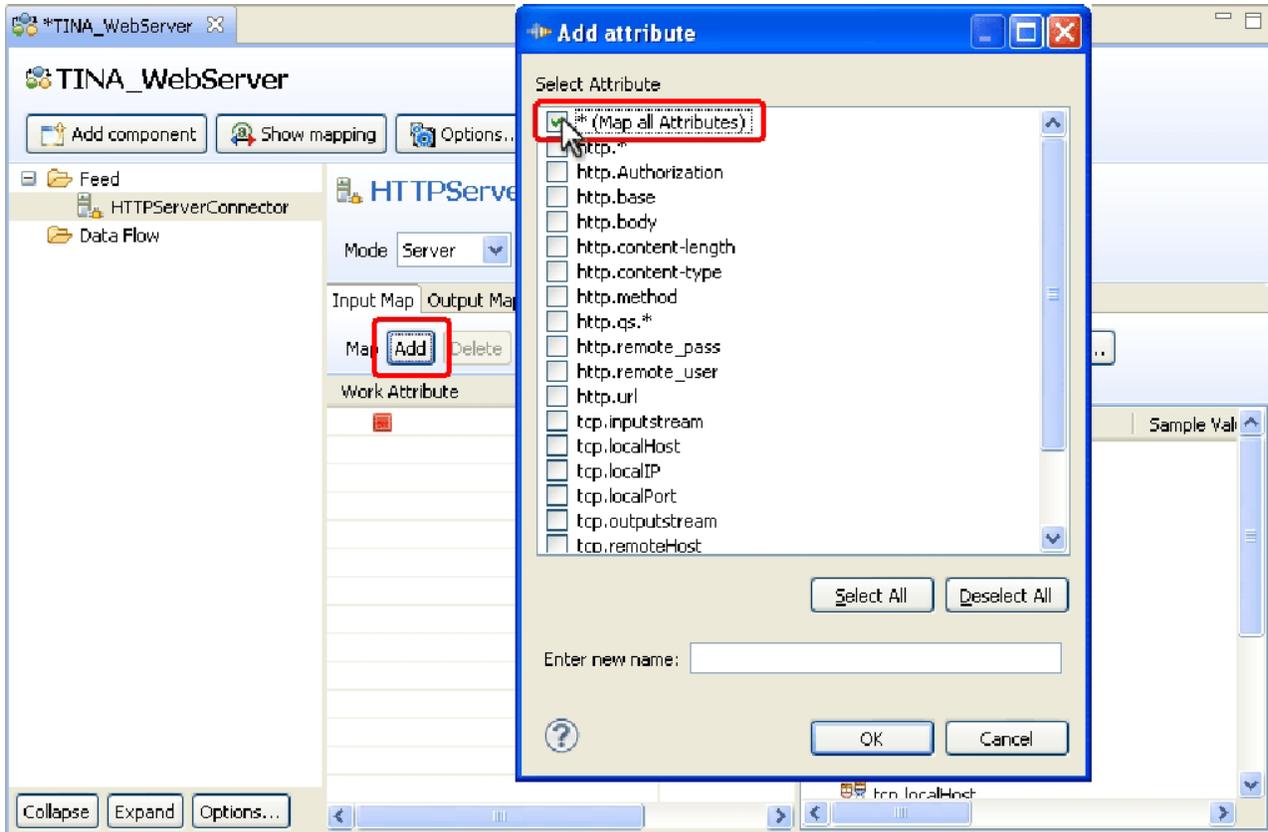


Figura 80. Incluir Item do Mapa de Atributos de Entrada

Basta selecionar a primeira opção no diálogo apresentado. Observe que você também poderia ter digitado o caractere de asterisco (*) no campo Entrada de novo nome. Esta é a regra especial de mapa de curingas que indica ao IBM Security Directory Integrator para mapear em todos os Atributos lidos pelo Conector.

Seu Mapa de Entrada será semelhante a este:

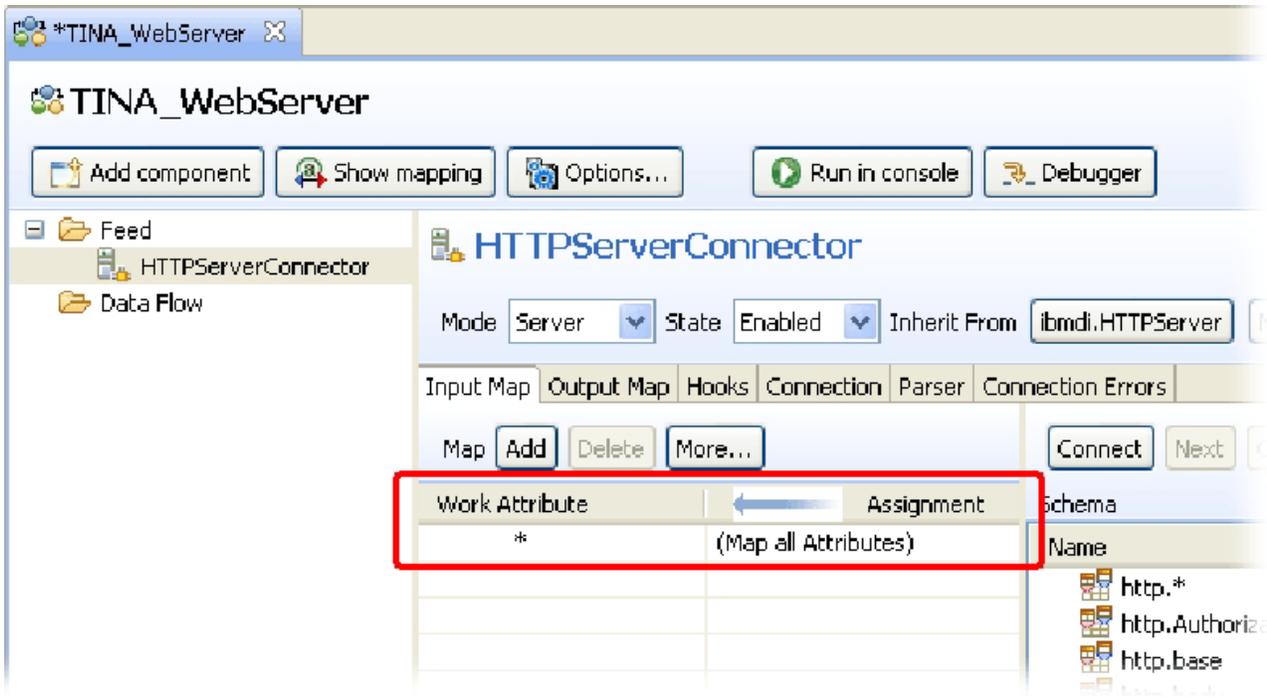


Figura 81. Item do Mapa de Curingas

Inclua também um item do mapa de curingas no Mapa de Saída para assegurar que quaisquer Atributos configurados na Entrada de Trabalho para a mensagem de resposta sejam mapeados de volta para o cliente.

Para testar esse componente, coloque um componente de Script 'Dump de Entrada de Trabalho' na seção Fluxo e, em seguida, **Execute** o AssemblyLine. A saída do log deve exibir a mensagem informando que seu Conector de Servidor HTTP está atendendo conexões HTTP na porta 80³⁴. Isso significa que seu AL está aguardando os clientes se conectarem, o que você faz agora abrindo um navegador e navegando para a seguinte URL:

`http://localhost:80`

Agora, examine sua saída de log. Aqui, você verá várias propriedades de cabeçalho TCP e HTTP que foram retornadas como Atributos.

34. Se por algum motivo essa porta já estiver em uso, simplesmente abra o painel Configuração para o Conector de Servidor HTTP e escolha uma outra porta.

```

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. Propriedades de Cabeçalho TCP e HTTP Retornadas como Atributos

O único Atributo que será de seu interesse para este exercício é o 'http.base', que contém essa parte da URL que aparece após o host e o soquete. Especificamente, você procurará por ele para conter o texto 'RunAL'.

Para verificar esse texto, inclua uma ramificação IF na seção Fluxo de Dados de seu AL. Denomine-o como 'RunAL detectado' e configure a Condição como: http.base contém 'RunAL'.

Se essa Condição de ramificação for avaliada como *verdadeira*, você desejará ativar seu AL 'CSV2XML_LookupMode'. Para fazer isso, você reutilizará o AssemblyLine Function Component a partir de seu AL 'Planejador', arrastando-o primeiramente para **Recursos > Funções** no painel Navegador e, depois, de volta para esse AssemblyLine, soltando-o em cima da nova ramificação IF.

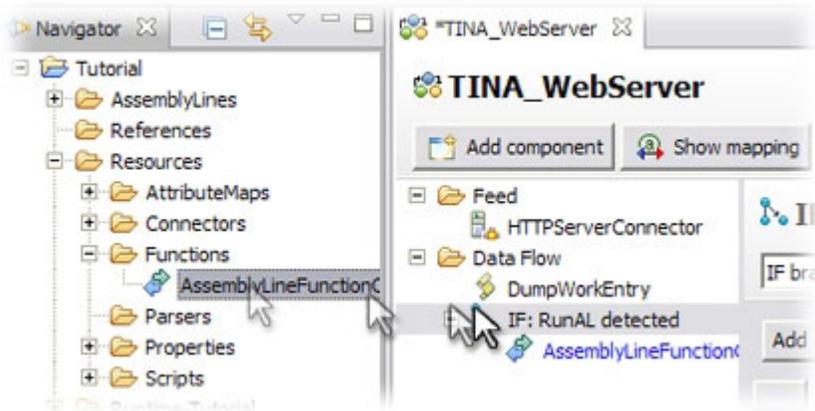


Figura 83. Arrastar no AssemblyLine Function Component (AL FC)

Agora, execute novamente seu AssemblyLine e insira este texto no campo de endereço de seu navegador:

`http://localhost/RunAL`

Agora, você verá o dump de Entrada de Trabalho na saída do log, seguido pelas estatísticas do AssemblyLine chamado.

```

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. Dump de Entrada de Trabalho Seguido por Estatísticas do AL

Seu serviço está funcionando, mas o exercício não termina aqui. Primeiramente, você o tornará um pouco mais atraente e muito mais fácil de ser usado, permitindo que seu AssemblyLine de servidor da Web retorne algumas páginas HTML. Isso normalmente requer uma ponta razoável de script. Felizmente você recebeu alguns arquivos de tutorial adicionais que tornam isso um simples caso de arrastar e soltar.

Antes de iniciar, desative o Script 'Dump de Entrada de Trabalho' para minimizar a saída do log. Em seguida, inclua uma ramificação ELSE imediatamente após 'IF RunAL detectado'. Nomeie-o como 'Return web page'. Agora, use um navegador de arquivos para localizar o arquivo de script com o nome "Return web page.script" no diretório Tutorial e arraste-o para **Recursos > Scripts**. Nesse local, você pode efetuar pull de seu AL, soltando-o em cima da ramificação ELSE. Agora, seu AL deverá ser semelhante a este:

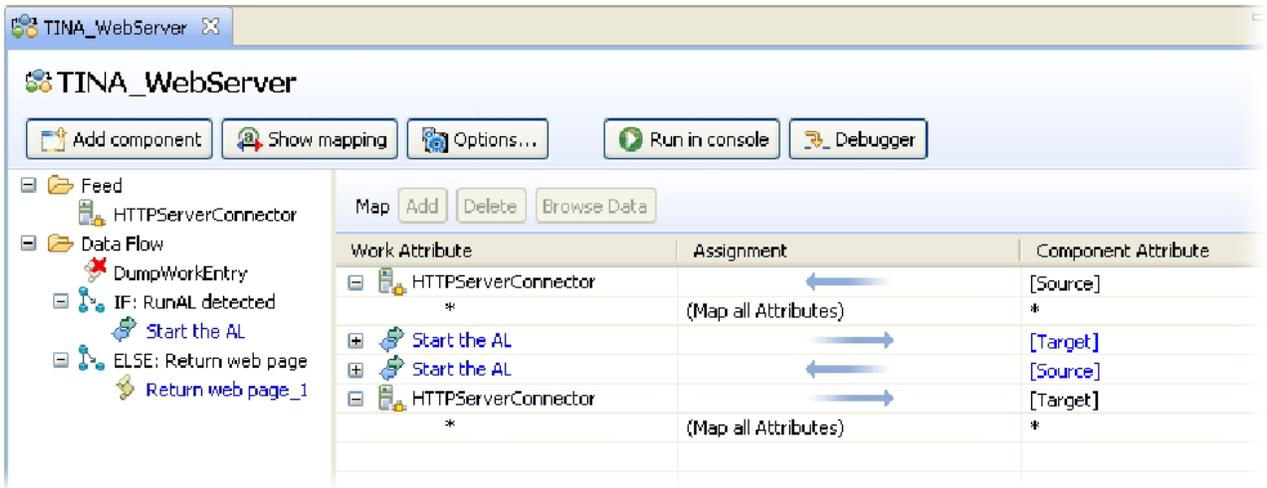


Figura 85. TINA_WebServer AssemblyLine Concluído

Execute o AssemblyLine novamente e quando discar `http://localhost`, você deverá ver esta página da Web:

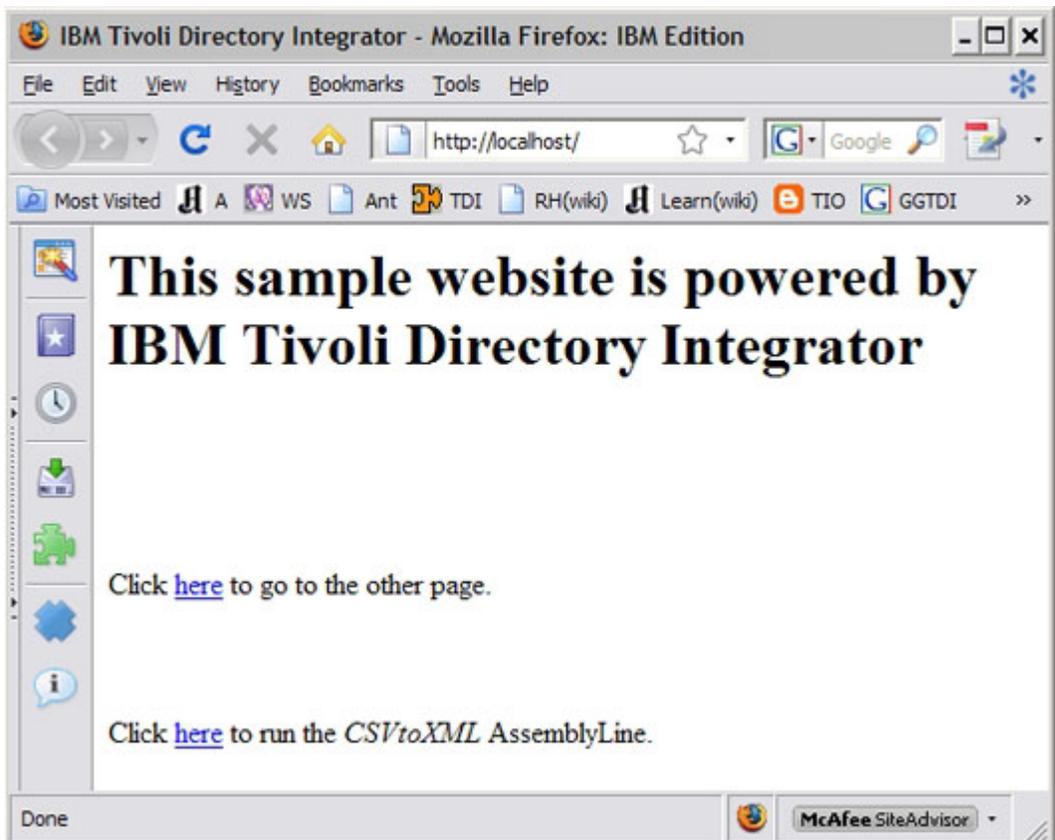


Figura 86. Interface da Web Simples para sua Solução

O link na parte superior navegará para `OtherPage.html`, enquanto o link na parte inferior deverá enviar o texto 'RunAL' necessário no `http.base` para ativar seu AssemblyLine.

E isso conclui a parte prática deste guia.

Capítulo 4. Fortalecendo suas Soluções de Integração

Conforme você observou, a criação de AssemblyLines pode ser um negócio muito rápido. Entretanto, uma execução de AL concluída não significa que sua solução está pronta para o 'momento principal'. Mesmo as tarefas de integração simples garantem um mínimo de análise de antecipação e resultado.

Algumas questões importantes para reflexão são:

- Todos os dados de origem estão sendo processados conforme esperado? Como você pode confirmar isto?
- As anomalias no conteúdo de dados e/ou qualidade foram detectadas? Elas estão tratadas?
- O processamento afeta outros conjuntos de dados, sistemas ou ALs? Como?
- A integração suporta uma sobrecarga de auditoria?
- Quem implementará sua solução? Quem a usará e quem a administrará, e como?

Para AssemblyLines de longa execução, como aqueles usados para sincronizações e para serviços de energia, você pode incluir considerações adicionais, como disponibilidade e tolerância a falhas, desempenho, escalabilidade e segurança.

O objetivo desta última seção é conscientizá-lo desses problemas, além de vários recursos e técnicas do IBM Security Directory Integrator que podem ser usados para tratar deles.

Nota: Se você for novo no IBM Security Directory Integrator, não se preocupe se esta seção parecer complexa e difícil de ser seguida. Em vez disso, volte e leia novamente estas seções à medida que sua experiência e conforto com o sistema aumentarem.

Legibilidade, Reutilização e Capacidade de Configuração

Todo trabalho de desenvolvimento requer resolução de problemas, manutenção e extensão. As soluções do IBM Security Directory Integrator não são uma exceção.

Você pode facilitar isso seguindo algumas diretrizes básicas.

1. Grave seus AssemblyLines tendo em mente que outras pessoas precisam entender, usar e mantê-los. Isso significa manter os ALs mais curtos possíveis e nomear os componentes de forma clara e descritiva. A lógica implementada no fluxo do AL por meio das Ramificações e Loops será mais simples para os não-programadores lerem e depurarem do que o Script 'oculto' em Ganchos ou compactado em componentes de Script.
2. Uma inferência à regra *Encurtar AL* é manter os trechos de script mais curtos possíveis. Em vez de gravar blocos monolíticos de código, divida-os em unidades menores, colocando-os ainda em componentes de Script separados para aumentar a legibilidade e a capacidade de depuração. É possível, então, desativar um SC para ignorar o código.

Uma outra maneira de melhorar a legibilidade e evitar a duplicação de código é usando a herança de componentes de Script (da pasta 'Scripts' na visualização em árvore do Navegador) e definindo funções para tarefas comuns. Um AssemblyLine é executado no contexto de seu próprio mecanismo de Script,

portanto, todas as variáveis e funções declaradas em um local estão disponíveis em toda parte. Um local comum para defini-las é nos Ganchos de Prólogo do AL ou em Scripts que foram selecionados como "Prólogos Adicionais"³⁵.

3. Escolha legibilidade acima da elegância quando chegar a seus algoritmos, tendo em mente que quando você pegar seu trabalho daqui a seis meses, provavelmente terá a impressão de que ele é de uma outra pessoa. A consideração com os colegas será também gentileza consigo mesmo.
4. Tenha em mente que as pessoas sem nenhuma qualificação para o Configuration Editor podem precisar modificar configurações e executar seus AssemblyLines. Os ALs podem ser facilmente iniciados a partir da linha de comandos:

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

Isto significa que você pode preparar scripts ou arquivos em log para facilitar isso.

5. Torne seus ALs mais simples de serem reconfigurados, externalizando as configurações dos parâmetros usando Propriedades. As propriedades são pares de valores das chaves que podem ser armazenados em arquivos ou bancos de dados e permitirão que sua solução seja reconfigurada fora do Config Editor. As propriedades são ligadas a parâmetros de componentes, clicando no rótulo do parâmetro e pressionando **Incluir Propriedade**.

As propriedades também podem ser consultadas e modificadas a partir de seus scripts com as chamadas `system.getTDIProperty()` e `system.setTDIProperty()`, permitindo também que você torne a lógica customizada facilmente alternável por meio de configurações de propriedades externas.

Além disso, as propriedades podem ser alteradas em um Servidor em execução usando o utilitário de linha de comandos, `bin/tdisrvctl`, que também permite iniciar e parar AssemblyLines, consultar o status e (re)carregar Configurações – tudo sem parar o Servidor.

6. Conforme já mencionado, o uso de caminhos relativos para arquivos torna mais fácil mover sua solução para uma nova instalação. Recomenda-se que você torne seus caminhos relativos ao diretório a partir do qual o arquivo XML de configuração é carregado. Isso pode ser acessado por meio da propriedade `{config.$directory}`, que pode ser usada para parâmetros de caminho específicos usando a opção **Texto c/ Substituição**; por exemplo:

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

7. Conforme já mencionado, mas especialmente ao construir soluções que serão implementadas e executadas por outros, não espere que esses usuários tenham qualificações do IBM Security Directory Integrator. Forneça arquivos em `lote/scripts` para iniciar seus AssemblyLines, incluindo ALs de teste e validação. Estes poderiam, por exemplo, simplesmente se conectar às origens de dados e relatar de volta o sucesso ou falha. Observe que para imprimir mensagens na linha de comandos do console para que seus arquivos em `lote/scripts` retornem informações de status, use o método do Servidor, `main.logmsg()`, em vez da versão do AL que você usou nos exercícios do tutorial: `task.logmsg()`. Esta última chamada enviará somente sua mensagem para o log.

Estas são apenas algumas considerações relevantes. É possível localizar mais em outras publicações do IBM Security Directory Integrator e nos grupos de notícias.

35. Os Prólogos Adicionais são executados antes de qualquer um de seus próprios Ganchos de Prólogo do AssemblyLine serem chamados. Eles são selecionados no painel Configurações do AssemblyLine, que pode ser acessado clicando com o botão direito do mouse em um AL e selecionando **Configurações do AssemblyLine...**

Criação de Log e Auditoria

O IBM Security Directory Integrator usa o *log4j* para fornecer gerenciamento de log flexível. É possível escolher entre vários *Anexadores* padrão, inclusive para syslog do Unix, log de eventos do Windows, arquivos diários e logs de rolagem. Os novos Anexadores também podem ser criados ou transferidos por download e usados.

Por padrão, somente a criação de log mínima do Servidor é ativada. No mínimo, você deve definir a criação de log para seus AssemblyLines usando o Anexador FileRoller, gravando na subpasta 'logs' de seu Diretório de Solução e nomeando o arquivo de log igual ao AssemblyLine. Portanto, para seu AL 'CSV2XML', você definiria um conjunto de arquivos de log de rolagem com base neste caminho de arquivo: logs/CSV2XML.log.

Observe que o método `logmsg()` permite definir opcionalmente o nível de log para sua mensagem, passando uma das seguintes palavras-chave como o primeiro argumento, logo antes de sua mensagem de log: DEBUG, INFO, WARN, ERROR, FATAL. Os níveis de log são inclusivos, portanto WARN incluirá ERROR e FATAL, DEBUG significa que mensagens de todos os níveis são registradas. Por exemplo, uma mensagem como:

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

será emitida somente por Anexadores configurados para a criação de log no nível DEBUG.

Você pode incluir mensagens de auditoria em sua solução, que podem ser ativadas e desativadas fora de um Servidor em execução, prefixando chamadas para as chamadas `task.logmsg()` com uma instrução IF que verifica o valor de uma propriedade. Por exemplo, este trecho de script pode aparecer em um Gancho 'DataFlow - Atualização Bem-sucedida':

```
if (system.getTDIProperty("MyProps","audit").equalsIgnoreCase("true"))
    task.logmsg("DEBUG", "Foram atualizados os seguintes dados: " + conn);
```

Usando o utilitário de linha de comandos `tdisrvctl` para alterar o valor da propriedade "audit" no Armazenamento de Propriedades "MyProps", é possível ativar ou desativar esse tipo de mensagem de auditoria para um Servidor em execução.

Em geral, é melhor registrar muitas informações do que poucas. Apesar disso, não convém inundar a saída do log. Pode ser difícil localizar mensagens de interesse em uma saída de log desordenada.

Problemas de Conectividade

Os problemas de conectividade podem geralmente ser divididos em duas categorias: erros de inicialização e conexões perdidas.

Por padrão, todos os componentes disparam conexões logo no início da operação AL durante sua fase de inicialização. Se por algum motivo uma conexão falhar neste ponto, o Gancho 'Prólogo – Em Erro de Conexão' será chamado, fornecendo um contêiner de script a partir do qual você enviará alertas ou até mesmo alterará configurações de parâmetros e tentará conectar-se novamente. De forma semelhante, se ocorrerem erros de conexão durante o ciclo do AL, você usará o Gancho 'DataFlow – Em Conexão Perdida' para tratar dessa situação.

Além dessa manipulação customizada, os componentes Connectors e Function também fornecem a funcionalidade integrada de *reconexão* por meio de uma guia **Erros de Conexão**. Aqui, você pode instruir o componente a tentar restabelecer uma conexão perdida ou, no caso de um problema de inicialização, a continuar tentando configurar a conexão.

No caso de erros de inicialização, não é uma prática comum ativar a reconexão a menos que você esteja tendo problemas recorrentes, como tempos limites durante a negociação de SSL ou problemas de conexão esporádicos semelhantes.

Entretanto, é recomendável ativar **Reconectar automaticamente na perda de conexão**, que permite que seu componente restabeleça sua conexão e, então, continue como se nada tivesse acontecido. Esteja ciente de que no caso de Conectores no modo *Agente Iterativo*, reconectar significará também que o 'cursor' de iteração poderá ser igualmente reconfigurado, de forma que o ciclo inicie novamente na primeira entrada do conjunto de resultados. Naturalmente, para Agentes Iterativos com reconhecimento de estado, como os Conectores de Detecção de Mudança, isto não é um problema, pois esses componentes usam as informações de estado automaticamente para retomarem do ponto onde pararam.

Disponibilidade do AssemblyLine

Melhorar a disponibilidade do AL significa duas coisas: 1) fazer o que for possível para assegurar que seus AssemblyLines não parem, e 2) reiniciar, o mais rápido possível, os ALs que pararam. Para cenários como migrações e sincronizações de longa execução, os AssemblyLines reiniciados também podem precisar continuar a partir de onde pararam no ponto de falha.

Um AssemblyLine parará se ocorrer uma exceção não-manipulada. É possível proteger-se contra essa eventualidade tratando de todos os erros, que em termos do IBM Security Directory Integrator equivale a ativar pelo menos o Gancho 'Padrão em Erro' de cada componente Connector e Function. Ativando os Ganchos de Erro, você está instruindo o Servidor a continuar apesar de uma exceção.

Conforme visto durante os exercícios do tutorial, se um AssemblyLine parar devido a um erro, você obterá um rastreamento de pilha precedido com informações sobre onde o erro ocorreu e o motivo do erro. Se você evitar que o AL pare ativando os Ganchos de Erro, então será sua responsabilidade relatar o status do erro usando os objetos especiais disponíveis para seus scripts.

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

O script de exemplo acima usa dois objetos desse tipo: `thisConnector` que sempre referencia o componente ao qual o script de execução está vinculado e a variável predefinida denominada `error`. O objeto `error` é uma Entrada, exatamente como `work` e `conn`, e contém Atributos como 'status', 'connectorname'³⁶ e 'message', além de outros detalhes relevantes sobre qualquer situação de erro recente. Como é um objeto Entry, você pode usar `task.dumpEntry(error)` para exibir seu conteúdo, além de referências diretas aos nomes de Atributos – por exemplo: `work.message` – ou simplesmente anexar `error` a uma mensagem de cadeia como no exemplo acima, pois todos os objetos Entry podem se converter em representações de cadeia, conforme necessário.

36. Você pode ter observado que a palavra 'connector' é às vezes sinônima de 'component', de forma que variáveis como `thisConnector` também podem se referir aos componentes de FC, de SC e AttributeMap. O mesmo se aplica ao Atributo 'connectorname' do objeto `error`, que também pode conter o nome de qualquer tipo de componente.

Para tratar de exceções que ocorrem em scripts, como em Designação do Mapa de Atributos, em Ganchos e em Componentes de Script, quebre seu código em blocos try-catch. Isso permite capturar exceções e permite que você mesmo lide com elas:

```
try {
    res = myLib.callToSomeFunctionThatMightFail();
} catch (excptn) {
    task.logmsg("A chamada falhou com erro: " + excptn);
}
```

Além de tratar dos erros, você desejará usar o recurso Reconexão Automática descrito na seção anterior. Isso evitará que o AssemblyLine falhe devido a problemas de conectividade temporários, como ser expirado por uma origem de dados ou por um firewall.

Se por algum motivo seu AssemblyLine ainda parar prematuramente, sua próxima etapa será reiniciá-lo. Uma abordagem é usar a ferramenta Web Admin para definir os comportamentos de falha/resposta.

Um outra abordagem também complementar é criar um AL 'Ativador' e alavancar o AssemblyLine Function Component usado nos exercícios do tutorial da última seção. Colocando esse AL FC em um ConditionalLoop que nunca para – ou seja, com uma Condição que é sempre *verdadeira* – e, depois, configurando o AL FC para chamar o AL de serviço desejado e aguardar sua conclusão, você assegura que sempre que o controle retornar ao 'LauncherAL', o Loop sem fim simplesmente reiniciará seu fluxo de dados novamente.

Se você aplicar a técnica de reinício de loop descrita acima para um AL de sincronização com base em um dos Conectores de Detecção de Mudança (ou o Mecanismo Delta, descritos em outra parte), o AssemblyLine continuará automaticamente a partir do ponto onde falhou. Se não, a manipulação do estado ficará sob sua responsabilidade. Uma técnica comum é usar o Armazenamento do Sistema para persistir informações de estado, como registros de data e hora ou outros valores de chave para conjuntos de resultados classificados. Por conseguinte, sempre que o AssemblyLine for inicializado, essas informações de estado serão aplicadas ao Conector de Agente Iterativo, a fim de retomar o processamento imediatamente após a entrada tratada anteriormente.

Se a origem de dados de iteração para o AL não suportar devoluções classificadas, poderá ser necessário iniciar a iteração novamente a partir do início. Neste caso, observe que o modo Atualização do Conector oferece um recurso **Computar Mudanças** que compara os Atributos Mapeados de Saída com aqueles localizados atualmente no sistema de destino, ignorando a operação de modificação se nenhuma diferença for detectada.

É possível evitar um único ponto de falha introduzindo um transporte seguro entre múltiplos Servidores IBM Security Directory Integrator, como o IBM MQ. Deste modo, qualquer número de Servidores (e AssemblyLines) pode ser usado para iniciar o processamento, colocando dados e até mesmo instruções de processamento na fila. Na extremidade de recebimento, múltiplos Servidores/ALs os selecionam por ordem de chegada e executam o trabalho solicitado. Isso não resulta somente em uma solução mais robusta, mas permite a escalação fácil por meio da inclusão de AssemblyLines do emissor e do receptor.

Esta foi uma discussão breve de um assunto muito mais amplo. Entretanto, o objetivo é mais para inspiração do que para a prescrição de uma abordagem

específica. Aconselhamos você a examinar os Web sites de comunidades e os grupos de discussão para obter recomendações e exemplos mais específicos.

Escala e Desempenho

Novamente, este é um tópico que merece uma discussão mais aprofundada do que você encontrará aqui. Entretanto, vale a pena mencionar alguns pontos, mesmo que seja de uma forma geral.

O fator de registro principal para a velocidade de seus fluxos de dados será E/S; o tempo que leva para recuperar dados das origens ou para efetuar push de mudanças para os destinos supera bastante o tempo de processamento em seu AL. Além disso, a negociação de conexões também pode ser muito cara, especialmente quando o handshake de segurança está envolvido. Ter isso em mente ao projetar e construir seus AssemblyLines impactará diretamente o desempenho deles.

Por exemplo, imagine um AL baseado no Modo de Servidor que recebe pedidos de protocolo de clientes, como a solução do Servidor HTTP que você construiu no último exercício. Cada pedido recebido faz com que um AssemblyLine de serviço seja ativado para executar o trabalho real. Se esse AL chamado precisar inicializar componentes, o tempo de ida e volta para cada pedido será pelo menos a soma de todos os tempos de conexão.

Há três maneiras de aliviar essa situação:

- Deixe que o AL no Modo de Servidor principal execute o processamento real em vez de despachá-lo, conseqüentemente não exigindo que as conexões sejam configuradas e fragmentadas para cada pedido;
- Projete o(s) AssemblyLine(s) de serviço para que possa(m) ser chamado(s) no *Modo Manual/Ciclo*. O modo Manual/Ciclo faz com que o AL inicialize quando o AL FC é inicializado. Além disso, o AssemblyLine Function Component aciona o AL de serviço somente um único ciclo para cada chamada. Dessa maneira, o AL de serviço age exatamente como um componente do AssemblyLine de chamada e deve ser construído para acomodar esse comportamento;
- Use o Conjunto de Conectores Globais. Esse recurso é descrito no *Referência* e permite que você defina um conjunto de Conectores que são inicializados na inicialização do Servidor e compartilhados entre os AssemblyLines, conforme necessário.

Uma outra maneira de melhorar o desempenho é dividir as tarefas de processamento pesadas entre múltiplos AssemblyLines simultâneos. Pegue, por exemplo, uma tarefa de migração em que, em vez de ter um AssemblyLine trabalhando com o conjunto de dados de origem inteiro, você ativa múltiplos ALs, em que cada um deles manipula um subconjunto. Como você pode passar parâmetros de inicialização para um AssemblyLine ao chamá-lo, um único AL pode ser desenvolvido, o qual é iniciado múltiplas vezes com um parâmetro de filtro para controlar o intervalo de dados que a instância deve processar.

Ainda, uma outra técnica é incorporar um barramento de mensagem à sua solução, conforme descrito a seção anterior. Essa abordagem foi usada com grande sucesso por alguns dos maiores clientes da IBM.

Nos casos em que a velocidade de processamento é dificultada por links de rede instáveis ou sistemas com baixa disponibilidade, você pode implementar ALs adicionais como tarefas em segundo plano para sincronizar dados de difícil alcance

com os armazéns de alta velocidade locais. Em particular, ao implementar serviços em tempo real, esta técnica pode ajudar a assegurar um tempo de resposta satisfatório para pedidos do cliente.

Monitorando

Prontas para uso, suas soluções do IBM Security Directory Integrator são administradas de forma rápida e simples usando a ferramenta Web Admin. Esse plug-in do Integrated Service Console (ISC) é um aplicativo AppServer que pode monitorar qualquer número de soluções em execução em qualquer número de Servidores em sua infraestrutura. Além dos recursos prontos para uso para visualizar estatísticas do AL, logs e horários de início/parada, a ferramenta Web Admin permite customizar o console de funcionamento e incidente, além de definir planejamentos e comportamentos de falha/resposta para manter os AssemblyLines em andamento.

Também é possível configurar suas soluções do IBM Security Directory Integrator para enviar eventos de status, por exemplo, como traps SNMP ou usando o formato de evento customizado do sistema. O sistema está prontamente configurado para administração e monitoramento do JMX, por exemplo, usando o ITM.

O Depurador de AssemblyLine

Embora já mencionado, isto justifica uma repetição: O tempo que você gastar para obter qualificação com o Depurador de AssemblyLine será recompensado dez vezes mais no desenvolvimento, na resolução de problemas e na implementação aceleradas da solução.

Apêndice. Guia do EasyETL

'ETL' significa Extract, Transform e Load e se concentra em obter dados de um local, alterá-los conforme necessário e colocá-los em outro local. 'EasyETL' é um recurso do IBM Security Directory Integrator que permite executar essa operação de forma rápida e interativa com apenas alguns pressionamentos de tecla.

Exemplos comuns de ETL são:

- Exportando registros do banco de dados, Documentos do Notes, entradas de diretório ou, até mesmo, mensagem recebida ou mensagens MQ para um arquivo;
- Carregar dados do arquivo para um sistema ou armazenamento de dados;
- Migrar dados diretamente de um sistema para outro ou entre versões do mesmo sistema, no caso de atualizações de software/esquema.

O IBM Security Directory Integrator EasyETL permite resolver estes e outros cenários em algumas etapas intuitivas, resultando em soluções adequadas para necessidades de movimentação de dados exclusivos, bem como para fluxos de dados essenciais em sua infraestrutura.

A primeira etapa na criação de uma nova tarefa do EasyETL é escolher sua Origem de Entrada e selecionar os atributos que você deseja transferir. Já neste ponto, o IBM Security Directory Integrator permite que você execute sua tarefa do EasyETL e colete os dados lidos do buffer de cópia para colagem. Se os dados precisarem ser transformados ou calculados, o EasyETL permitirá incluir a transformação e executar novamente a tarefa ETL, bem como copiar/colar os dados transformados. E você também pode escolher um Destino de Saída e fazer com que a sua tarefa do EasyETL grave os dados diretamente nesse local.

Depois que a solução EasyETL estiver funcionando conforme desejado, o IBM Security Directory Integrator poderá gerar os recursos da linha de comandos (arquivos em lote ou scripts) para ativar e planejar a tarefa de integração. Finalmente, o EasyETL alavancará os recursos de detecção de mudança do IBM Security Directory Integrator para transformar rapidamente a tarefa ETL em uma tarefa de sincronização de dados.

Nota: A boa notícia para usuários do IBM Security Directory Integrator é que cada solução EasyETL é um projeto do IBM Security Directory Integrator com um único AssemblyLine e ele pode ser aberto em um ambiente de desenvolvimento completo. No entanto, depois que você alterá-lo, não estará mais disponível como EasyETL.

Usando o EasyETL

Ative o IBM Security Directory Integrator Config Editor e selecione a área de trabalho a ser usada. Quando o IBM Security Directory Integrator é iniciado pela primeira vez, ele é aberto na página de Boas-vindas³⁷.

³⁷. É possível retornar para a página de Boas-vindas a qualquer momento selecionando **Ajuda > Bem-vindo** no menu principal.

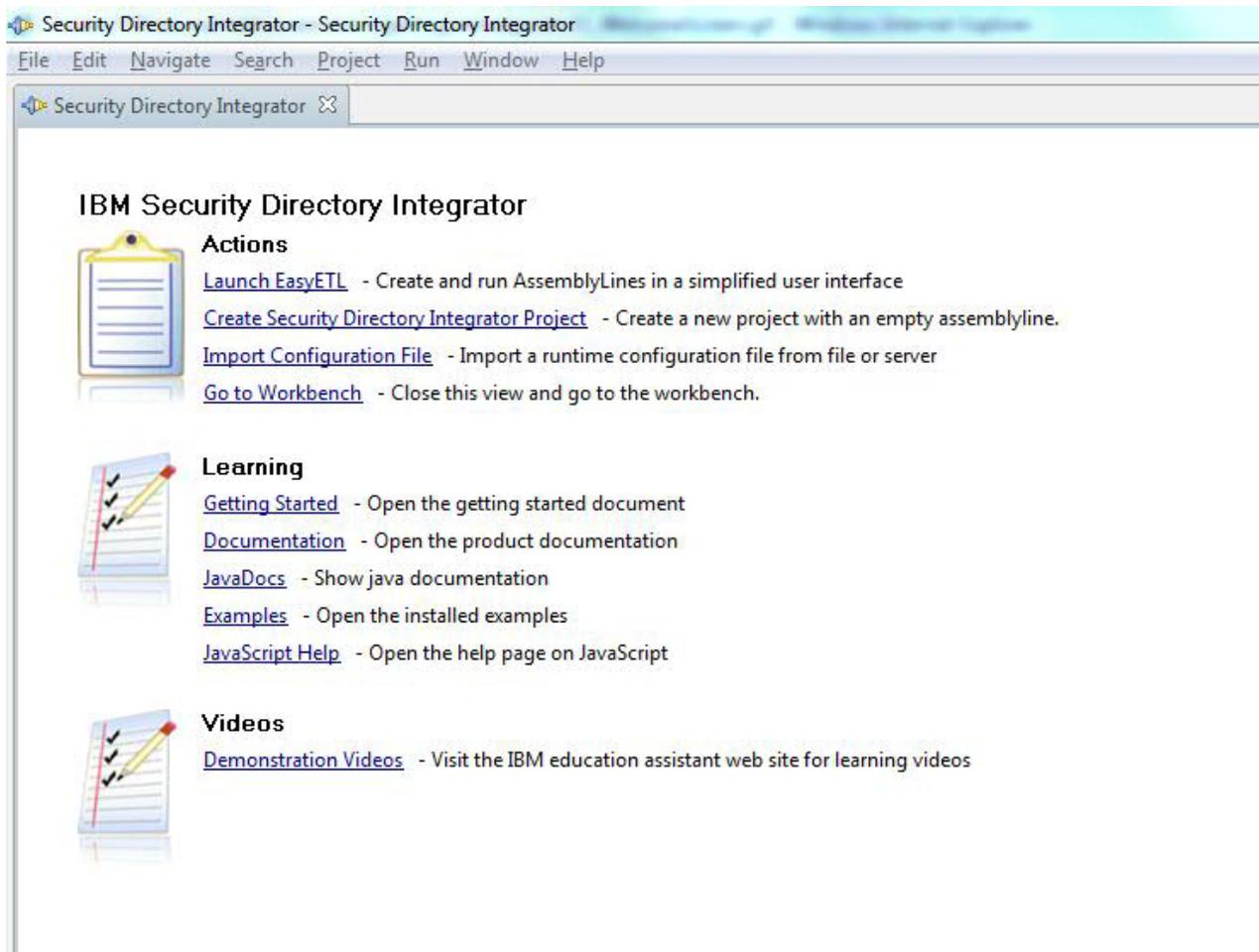


Figura 87. Tela de Boas-Vindas

Como mostrado na captura de tela anterior, o primeiro link aqui abre o ambiente de trabalho do EasyETL³⁸. Clique nesse link para abrir o ambiente de trabalho do Security Directory Integrator EasyETL.

38. EasyETL é uma *perspectiva* Security Directory Integrator e é possível alternar entre perspectivas usando a seleção de menu: **Windows > Abrir Perspectiva > Outra...** Se você tiver feito mudanças em uma perspectiva e desejar reconfigurá-la de volta para o padrão, basta selecionar **Windows > Reconfigurar Perspectiva**.

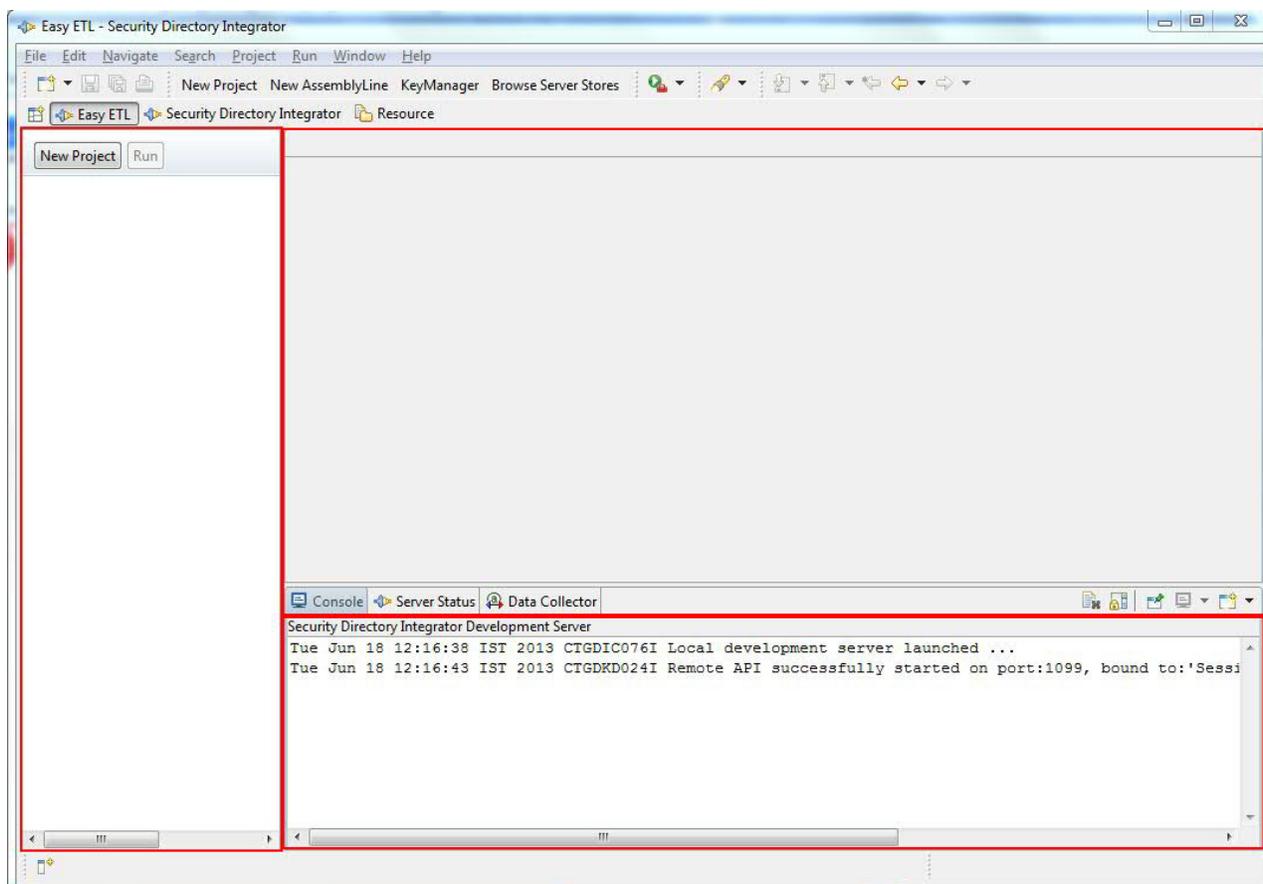


Figura 88. Figura 2. Ambiente de Trabalho do EasyETL

O ambiente de trabalho do EasyETL mostra três itens:

- O Navegador de Projeto que lista as tarefas ETL. Você pode clicar com o botão direito do mouse em qualquer Projeto para executá-lo ou para criar recursos de linha de comandos para ativá-lo;
- Um editor Simple AssemblyLine³⁹ para cada projeto aberto;
- Várias Visualizações como guias ao longo da parte inferior da tela. Por padrão, você obtém três Visualizações:
 - a saída do Console do Servidor IBM Security Directory Integrator de teste;
 - a visualização de status do Server, em que é possível monitorar o Server e todos os Projetos do EasyETL em execução;
 - e o Data Collector em que os dados resultantes de cada ciclo são exibidos em uma lista tabular.

Você localizará mais informações sobre cada seção à medida que percorrer este documento.

39. Um 'AssemblyLine' é a implementação de um fluxo de dados no IBM Security Directory Integrator. Sendo assim, quando você cria ou abre um Projeto do EasyETL, o AssemblyLine subjacente é apresentado no editor.

Observe também que o termo 'AssemblyLine' é abreviado como 'AL' nesta e em outra literatura do IBM Security Directory Integrator.

Criando um Projeto

Crie um novo Projeto EasyETL pressionando o botão **Novo Projeto** na parte superior do Navegador de Projetos.

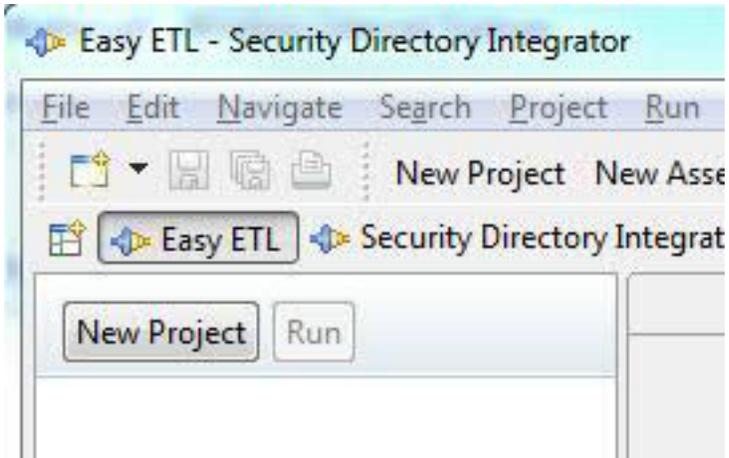


Figura 89. Botão Novo Projeto

Denomine este Projeto como 'CSVtoXML' e pressione **Concluir**. O seu novo Projeto é aberto no editor Simple AL.

Figura 90. Editor Simple AssemblyLine

O editor fornece duas listas suspensas: uma para selecionar a origem de Entrada e uma para o destino. A área abaixo fica vazia (com exceção do texto de assistência) até você escolher sua origem.

Configurando entrada para seu ETL AL

Configure a entrada para seu EasyETL AssemblyLine ao clicar no menu suspenso das Informações de Origem e selecionar **Conector de Arquivo**.

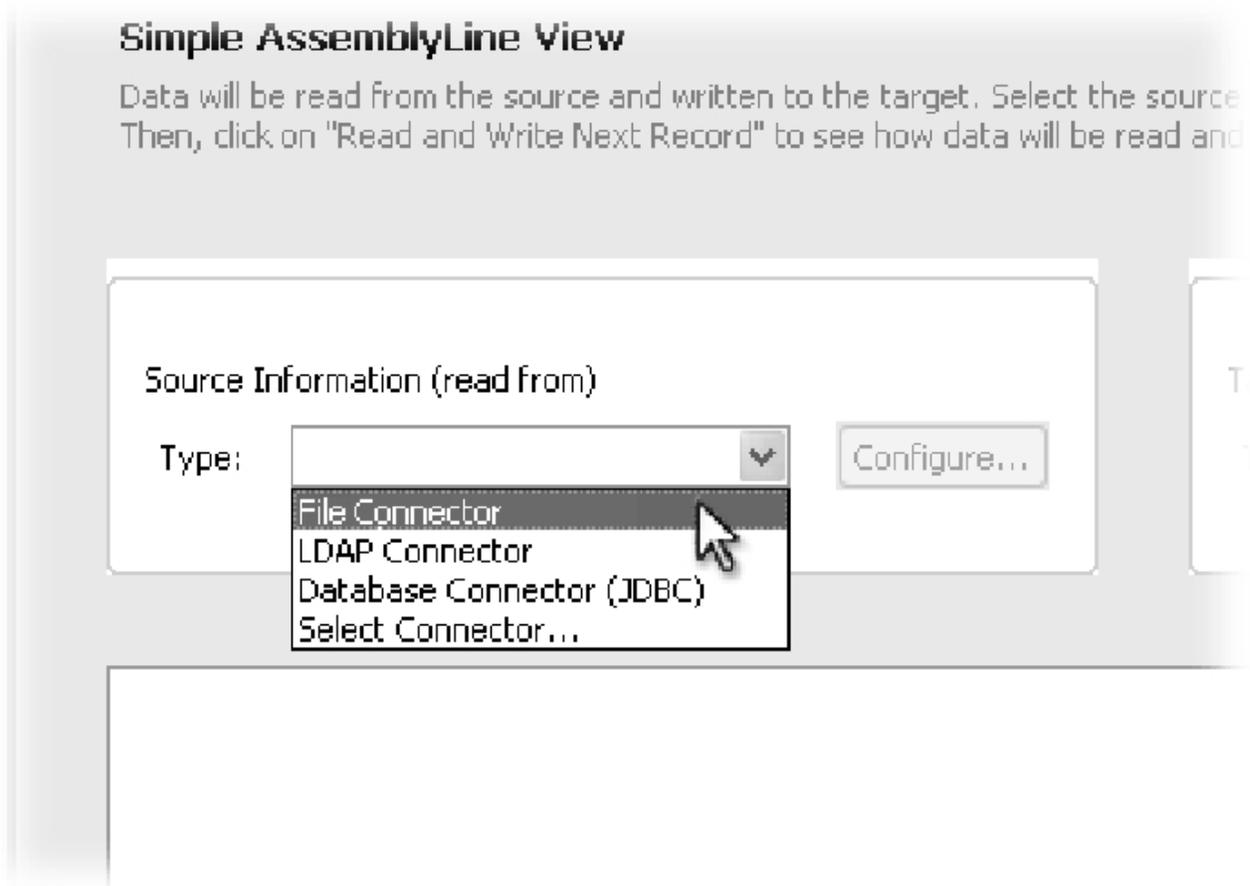


Figura 91. Selecionando Informações de Origem

Em seguida, você verá o diálogo de configuração deste Conector.

Aponte para o parâmetro Caminho de Arquivo no arquivo 'People.csv' localizado aqui:

```
TDI_HOME/examples/Tutorial
```

em que *TDI_HOME* é substituído pelo diretório de instalação do IBM Security Directory Integrator em sua máquina⁴⁰.

40. Note que o IBM Security Directory Integrator suporta a barra e a barra invertida como o separador de caminho durante a execução no Windows. Suas soluções do IBM Security Directory Integrator serão mais móveis entre o Windows e todas as outras plataformas em que o IBM Security Directory Integrator é executado, se você usar a barra.

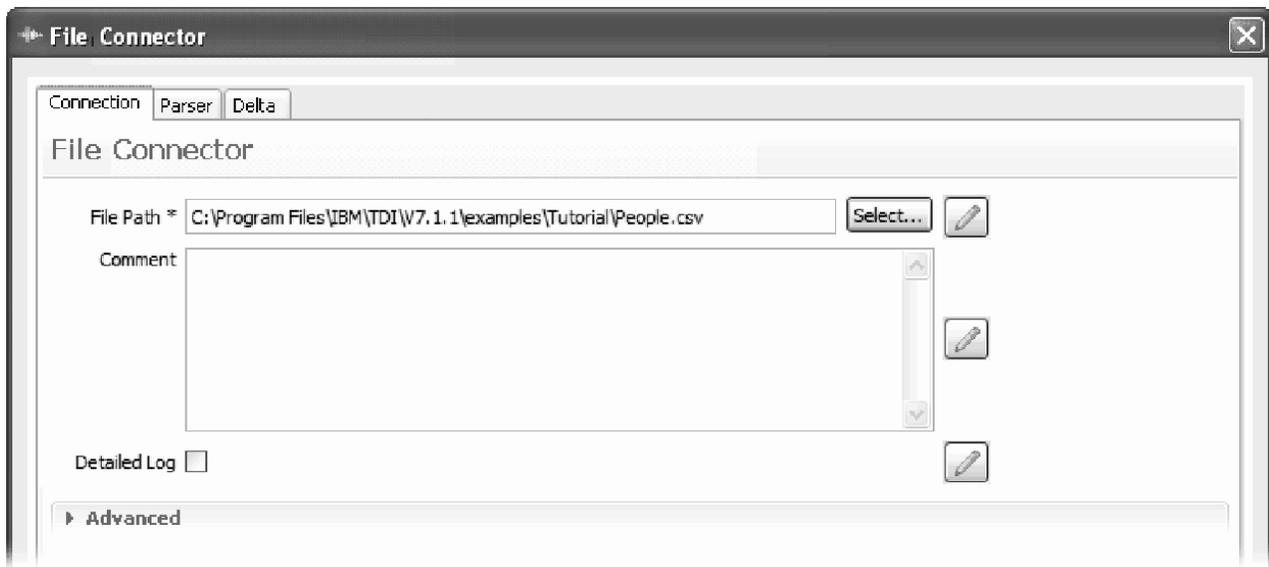


Figura 92. Configurando o Parâmetro Caminho de Arquivo

Agora, clique na guia com rótulo **Analizador** e selecione o **Analizador CSV**, mantendo os parâmetros de configuração no estado em que se encontram. Finalmente, pressione o botão **Conectar** na parte inferior da caixa de diálogo para testar a conexão e descobrir os Atributos disponíveis.

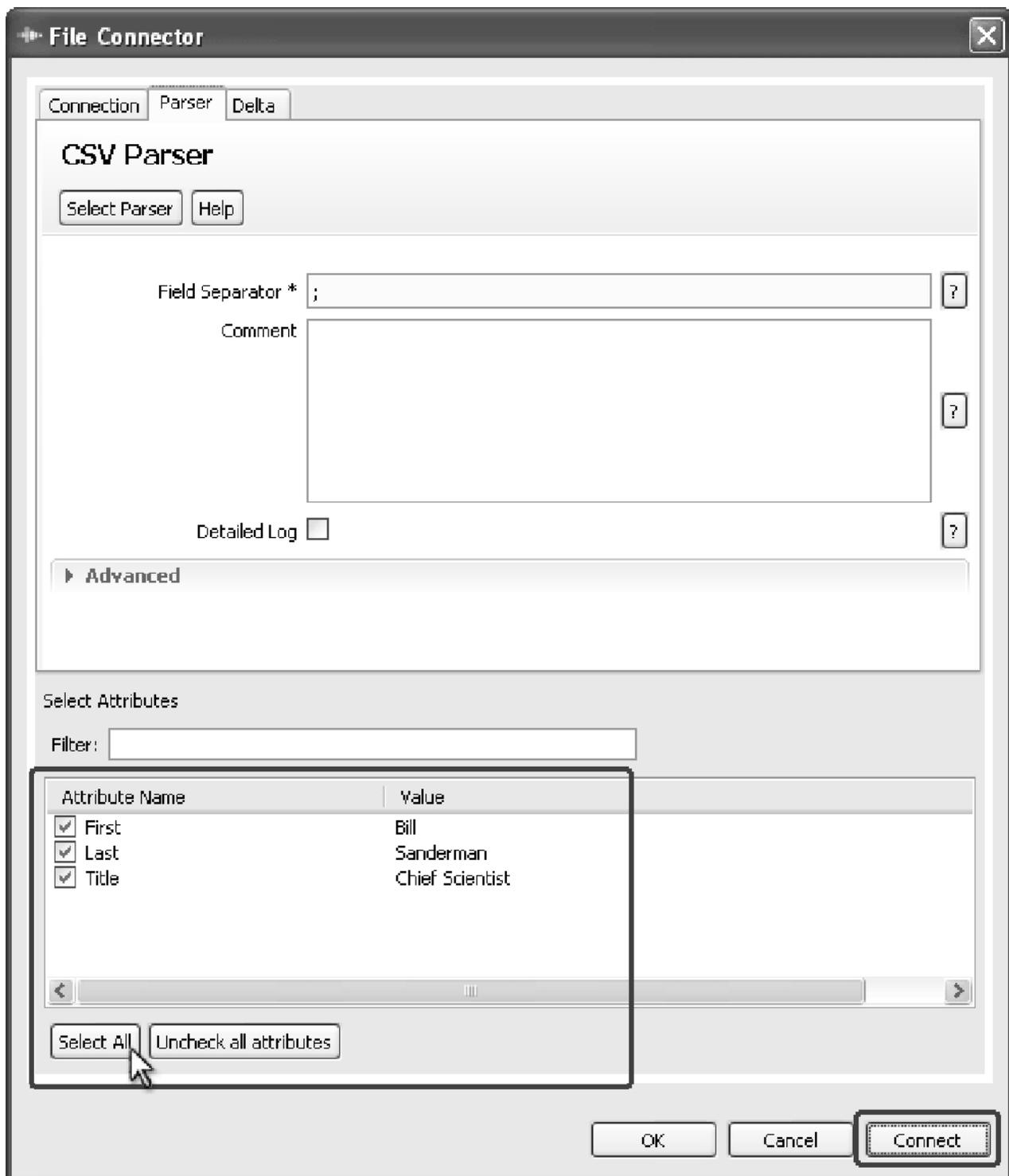


Figura 93. Testando a Conexão e Descobrimdo o Esquema

O esquema para o sistema conectado é exibido e, a partir desse ponto, você pode escolher qual destes usar em seu fluxo de dados. Para este exemplo, use o botão **Selecionar Tudo** e pressione **OK** para fechar o diálogo de configuração.

De volta no ambiente de trabalho do EasyETL, você verá que a metade inferior do editor Simple AL foi alterada para indicar que agora você tem uma Origem de

Entrada configurada. O IBM Security Directory Integrator agora apresenta um botão para percorrer um registro dessas informações de cada vez, bem como botões para executar a tarefa ETL até a conclusão e pará-la.

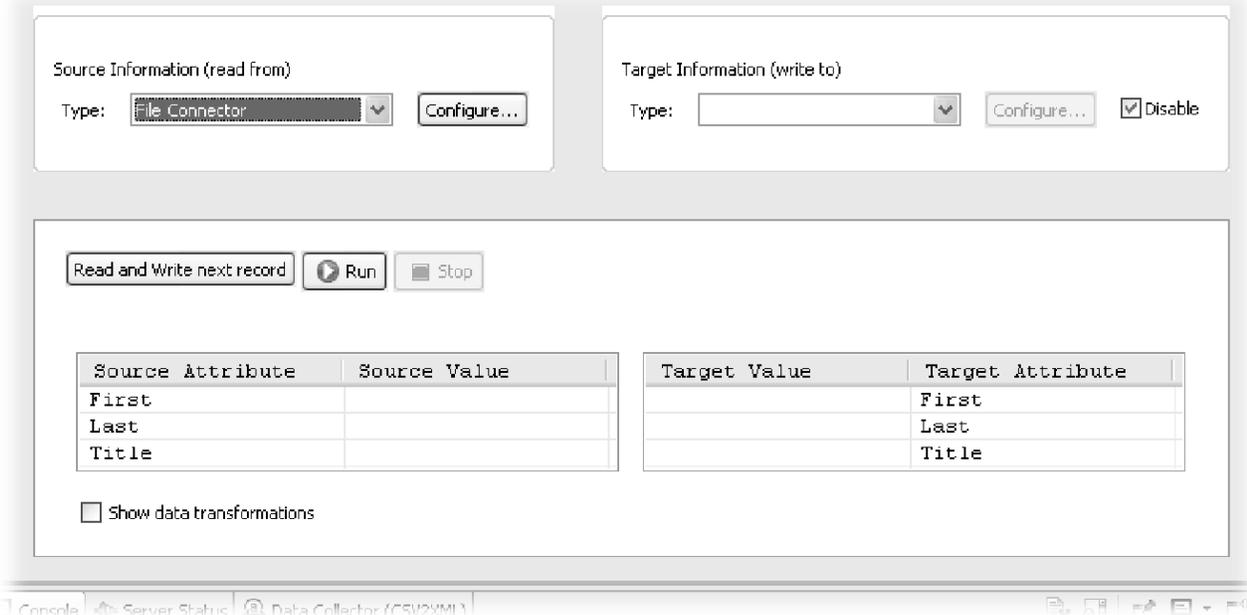


Figura 94. Origem de Entrada Configurada

Abaixo desses botões estão duas caixas em grade chamadas *Visualizador de Dados* que listam os Atributos tratados por seu fluxo de dados. O Visualizador de Dados à esquerda mostra seus Atributos de entrada. Aqueles que você selecionou para leitura na etapa anterior são exibidos em negrito na parte superior e são chamados *Mapa de Entrada*. Abaixo serão exibidos todos os Atributos desmarcados em cinza e você pode incluí-los para mapeamento de entrada dando um clique duplo neles. De maneira similar, remova os Atributos do mapa dando um clique duplo neles ou excluindo-os⁴¹.

A caixa à direita é o Visualizador de Saída, que mostra o conjunto de Atributos a serem gravados. Nos termos do IBM Security Directory Integrator, este é o *Mapa de Saída* e, por padrão, é idêntico à lista que você selecionou para o seu Mapa de Entrada. Note que é possível alterar o nome de qualquer Atributo de Saída clicando na coluna da direita e, em seguida, editando o valor. Use esta técnica para renomear o Atributo 'Title' para 'JobTitle'.

41. Também é possível clicar com o botão direito do mouse e escolher **Incluir Atributo** ou **Remover Atributo**.

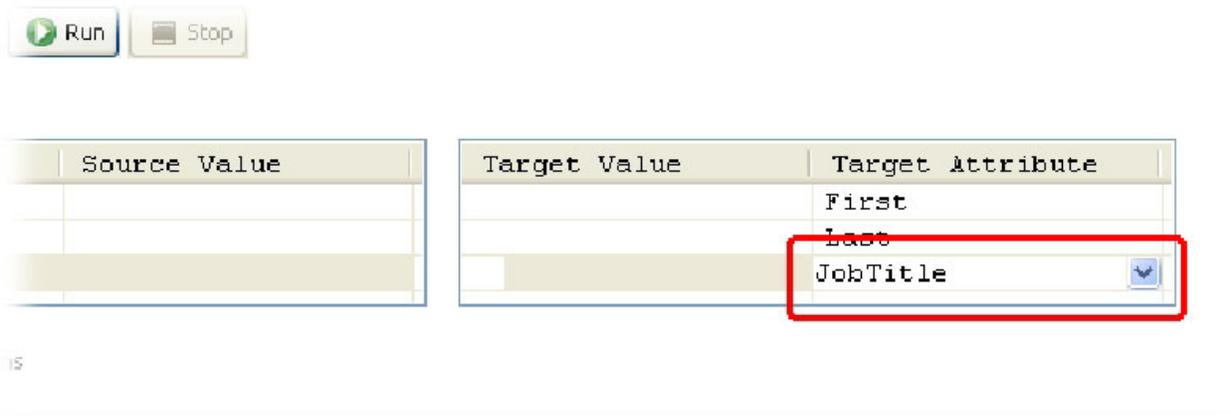


Figura 95. Renomeando um Atributo de Saída

O seu projeto do EasyETL agora está pronto para o primeiro teste.

Executando o EasyETL AssemblyLine

Selecione a visualização **Coletor de Dados** na parte inferior da tela e pressione o botão **Ler e Gravar Próximo Registro**. Isso faz com que ocorra o seguinte:

1. Haja um atraso, pois o EasyETL AssemblyLine é transferido para o Servidor em execução e iniciado;
2. O primeiro registro é lido e analisado desde sua origem de entrada CSV e os dados são exibidos nas telas em grade de entrada e saída;
3. Os Atributos selecionados para saída são gravados e coletados na visualização Coletor de Dados⁴².

Dessa forma, embora você não tenha selecionado um destino de Saída, ainda poderá executar e testar o seu projeto ETL, visualizando os dados de acordo com o seu fluxo no AssemblyLine.

42. O botão mais à esquerda na parte superior da visualização **Coletor de Dados** abre um diálogo de configuração, em que você pode aumentar o tamanho do **Buffer do Coletor de Dados**. Note que se você planejar coletar grandes quantias de dados, talvez seja necessário aumentar a memória disponível para o IBM Security Directory Integrator. Faça isso localizando o arquivo em lote/script 'ibmditk' na pasta de instalação do IBM Security Directory Integrator e abrindo-o em um editor. Próximo à parte inferior está a lista que ativa 'miadmin' e seria possível inserir o texto a seguir após a opção `-vmargs: -Xmx512M` Este exemplo permitirá que a memória de dados do IBM Security Directory Integrator seja aumentada para 512 Mbytes.

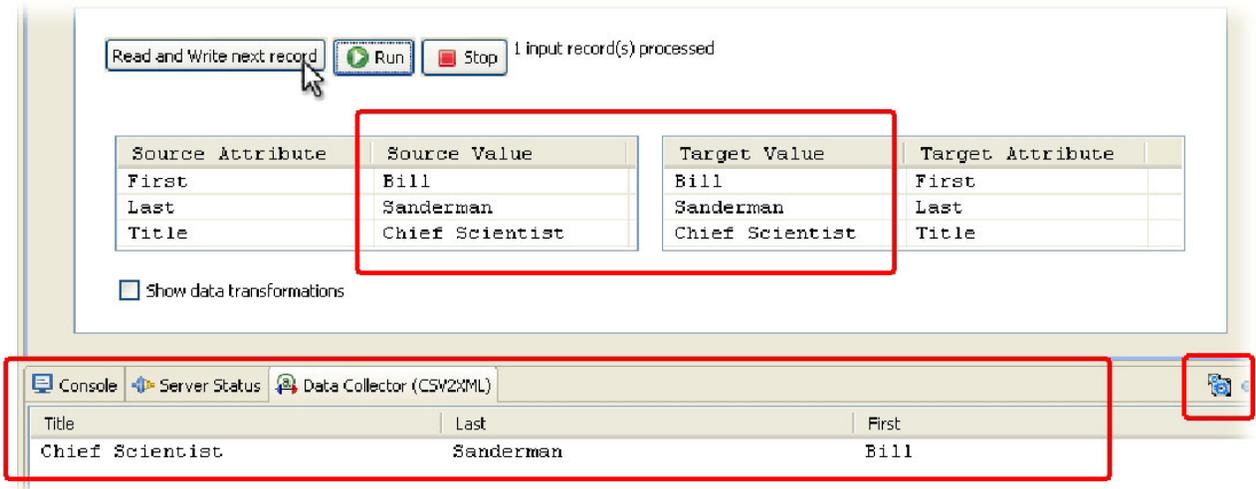


Figura 96. Um Registro Lido e Coletado

Toda vez que você pressiona o botão **Ler e Gravar...**, outro registro é lido, exibido e coletado. Se agora você pressiona o botão Executar, a tarefa do ETL será executada até a conclusão e você verá este relatório do AL.

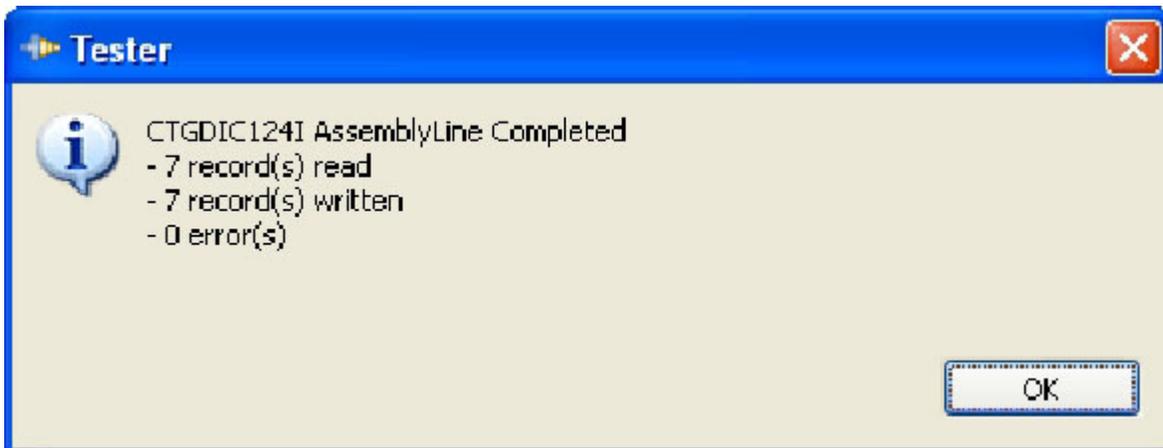


Figura 97. EasyETL AssemblyLine Concluído

Como mostrado no diálogo acima, nenhum registro foi realmente gravado em lugar nenhum. No entanto, o **Coletor de Dados** ainda fornece feedback visual prático sobre as informações que estão sendo extraídas e transferidas.

Além disso, você pode selecionar linhas nos dados coletados e copiar/colar essas informações em um arquivo ou em outro destino⁴³.

Transformações

Até agora, seus valores de Saída foram idênticos à sua entrada. No entanto, haverá situações em que você desejará manipular ou, até mesmo, calcular esses valores

43. Os dados são copiados em formato CSV para simplificar a importação de valores para planilhas e tabelas de relatório. 'CSV' significa Character Separated Value e o caractere separador usado pelo IBM Security Directory Integrator é o ponto e vírgula (;).

com base nos dados lidos. Isso pode ser feito no IBM Security Directory Integrator EasyETL gravando Transformações em JavaScript.

Para trabalhar com as Transformações, primeiro é necessário ativá-las ao marcar a caixa de seleção **Mostrar Transformações de Dados**.

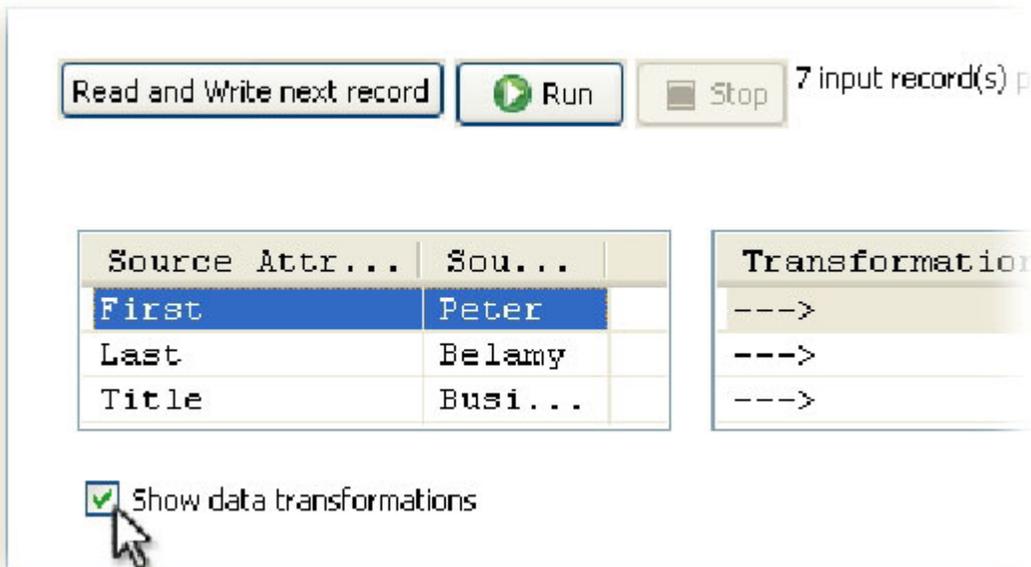


Figura 98. Ativando Transformação

Isso faz com que uma nova caixa em grade apareça entre os Visualizadores de Dados: o Visualizador de Transformação. Aqui você vê setas indicando que os três Atributos de Saída obtêm seus valores diretamente dos Atributos de Entrada – em outras palavras, sem transformações. Você definirá um novo Atributo de Saída e, em seguida, incluirá o script de Transformação para calcular o seu valor. Faça isso agora clicando com o botão direito do mouse no Mapa de Saída, escolhendo Incluir Atributo e denominando-o 'FullName'. Agora, dê um clique duplo na Transformação à esquerda desse novo Atributo de Saída e insira este fragmento de script:⁴⁴ `return First + " " + Last`

44. Note que você pode pressionar Ctrl + Espaço para obter uma lista de sugestões do que é possível digitar. Essa lista inclui alguns objetos especiais, como 'task' e 'main', e lista os Atributos que estão sendo lidos de sua Origem de Entrada na parte inferior.

Note também que a nota mostrada aqui para acessar Atributos de Entrada funciona apenas no EasyETL. No IBM Security Directory Integrator Workbench integral, você deve prefixar um nome de Atributo com o objeto Entry que o transporta - por exemplo, Work Entry (work). Como resultado, o exemplo acima deverá ser semelhante a este:

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

O Work Entry e outros conceitos do IBM Security Directory Integrator são discutidos na primeira seção deste documento.

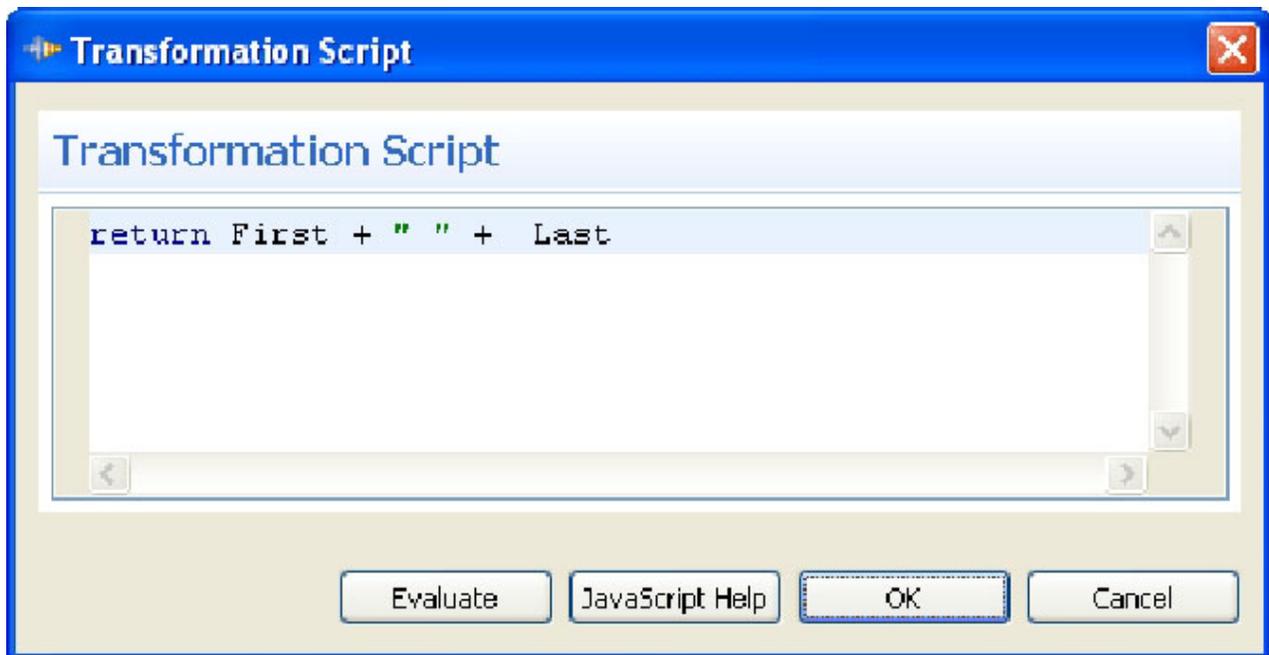


Figura 99. Mostrar Script de Transformação

Há um botão **Avaliar** no diálogo do editor Script de Transformação para testar o script, juntamente com um para ativar algumas dicas e alguns exemplos de JavaScript.

Pressione **Avaliar** agora para ter uma ideia de como sua Transformação funciona.

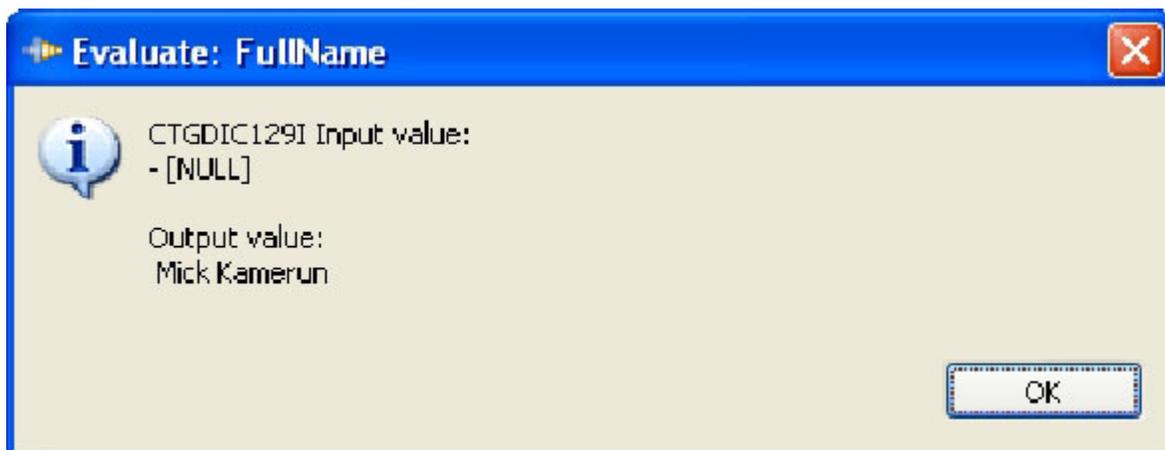
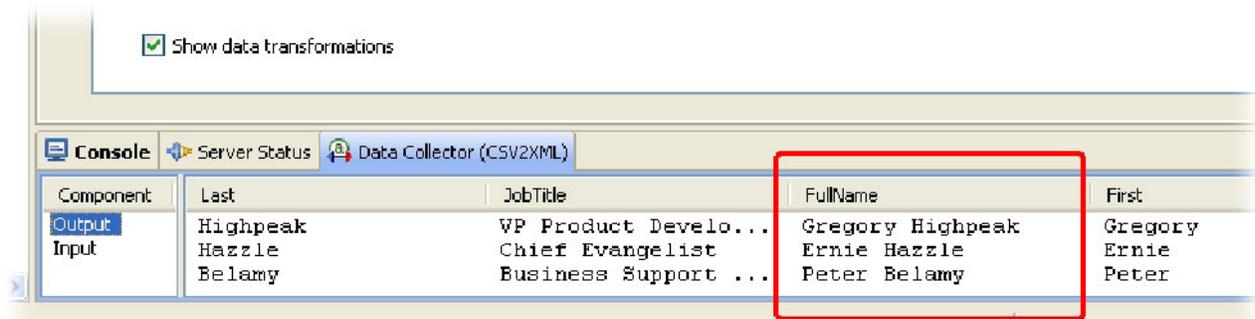


Figura 100. Avaliar Expressão

O valor de Saída mostrado foi calculado usando dados coletados quando você executou seu AssemblyLine. Feche o diálogo de resultados da avaliação pressionando **OK**.

Agora, pressione **OK** para controlar este script e feche o editor de Script de Transformação e execute novamente o EasyETL AssemblyLine pressionando **Executar** e, em seguida, visualizando as entradas coletadas. Observe como o **Coletor de Dados** agora fornece duas Coleções de Componentes para você escolher: *Saída* e *Entrada*. Escolha **Saída** e veja como seu script de Transformação

gerou um valor 'FullName' para cada entrada.



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. Coleta de Saída com o Atributo FullName Calculado

Portanto, agora você sabe como configurar uma Origem de Entrada e selecionar os Atributos a serem extraídos, além de como transformar esses dados para se ajustarem às suas necessidades. A próxima etapa é selecionar um destino de Saída e direcionar os dados nele.

Selecionando um destino de saída

Selecione **Conector de Arquivo** para seu destino de Saída usando o menu suspenso acima da Visualização de Saída e grave-o em um arquivo chamado 'Output.xml'⁴⁵. Escolha o **Analizador XML** e pressione **OK**. Note que você poderá usar o botão Connect para garantir que o caminho de arquivo inserido seja válido. No entanto, não haverá dados para serem descobertos, a menos que você aponte seu Conector em um arquivo XML existente.

Depois que nossa Saída estiver configurada, execute o seu ETL AssemblyLine novamente. Quando terminar, você poderá abrir o arquivo de saída e verificar os resultados.

45. Se você não inserir um caminho completo ou um caminho relativo, o IBM Security Directory Integrator baseará isso a partir do Diretório de Solução especificado durante a instalação.

```

- <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. Saída XML

Seu destino de Saída poderá ter sido facilmente uma tabela de banco de dados, assim como sua entrada poderá ser proveniente de um aplicativo Lotus Notes ou diretório LDAP. As etapas seguidas para criar este exemplo de tarefa ETL são as

mesmas, independentemente dos sistemas ou armazenamentos de dados com os quais você está trabalhando.

Detectando Mudanças

O IBM Security Directory Integrator fornece vários recursos para detectar mudanças nos dados de entrada. Além de oferecer um conjunto de Conectores de Detecção de Mudança, você também tem a opção de ativar o *Mecanismo Delta* para sua origem de Entrada.

O Mecanismo Delta usa capturas instantâneas de dados durante a leitura e as compara com as capturas instantâneas obtidas durante a execução anterior para determinar o que foi alterado. Estas entradas inalteradas são ignoradas e apenas as entradas modificadas são recuperadas para processamento no EasyETL AssemblyLine.

Pressione o botão **Configurar** para sua origem de Entrada e, em seguida, selecione a guia **Delta**.

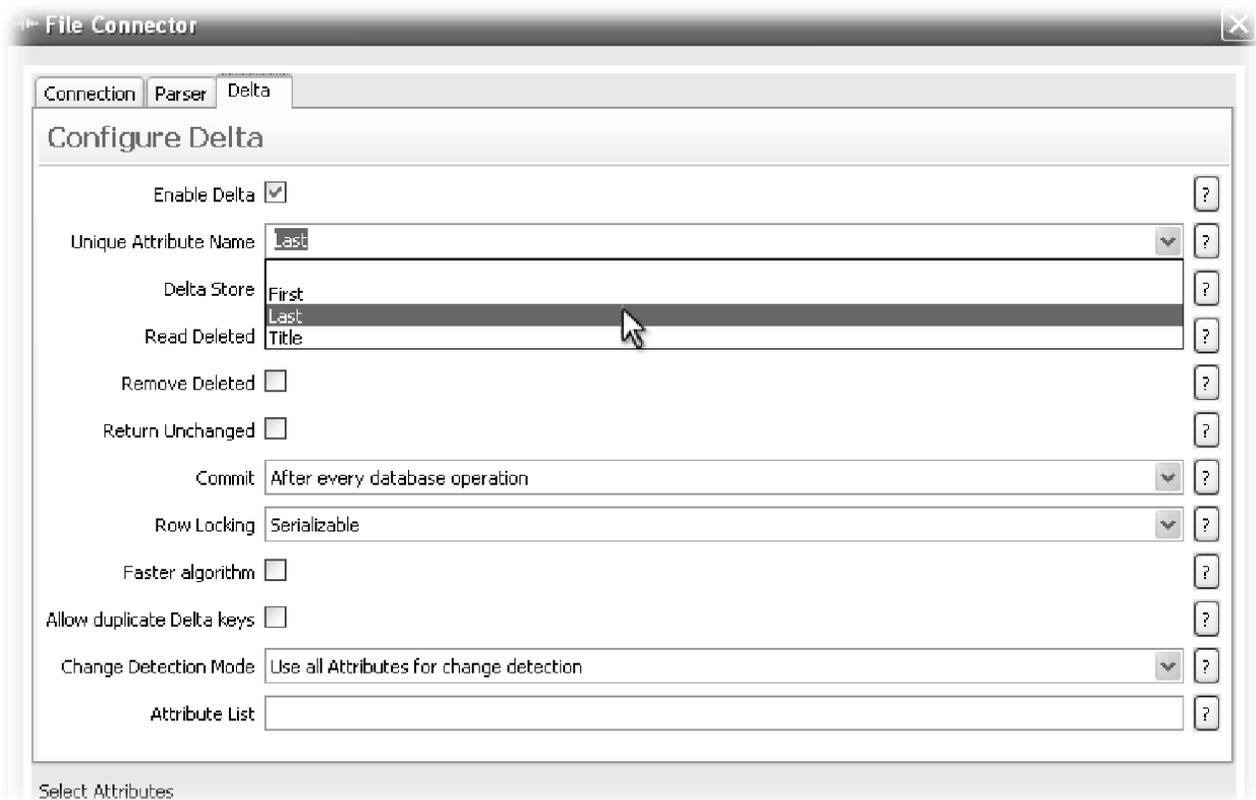


Figura 103. Configuração Delta

Você deve primeiro ativar o Mecanismo Delta ao marcar a caixa de seleção na parte superior do painel de configuração. Em seguida, use a lista suspensa para selecionar 'First' como o **Nome de Atributo Exclusivo**⁴⁶.

46. Como você pode ter deduzido, o Mecanismo Delta usa um de seus atributos de entrada para identificar exclusivamente capturas instantâneas. Se não houver um valor exclusivo disponível nos dados de entrada, você poderá especificar vários atributos que serão concatenados juntos a partir do ID da captura instantânea. Isso é executado digitando os nomes de vários atributos separados por um símbolo de mais (+). Por exemplo: First + Last

Existem vários outros parâmetros disponíveis aqui, alguns dos quais fazem mais sentido ao trabalhar no IBM Security Directory Integrator Workbench padrão e não no EasyETL. Por exemplo, embora um EasyETL AL possa detectar e transferir entradas novas e modificadas, ele não excluirá uma linha a partir de um banco de dados ou entrada em um diretório. No entanto, ele gravará essas informações em um destino de Saída, como um Conector de Arquivo com o Analisador LDIF. Os arquivos LDIF podem conter tags da operação de mudança e alguns sistemas suportam a importação LDIF.

É possível saber mais sobre os recursos integrais de Manipulação Delta do IBM Security Directory Integrator aqui:

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

Uma mudança que você pode desejar fazer é no parâmetro **Commit**. Isso controla quando capturas instantâneas novas e alteradas são confirmadas para o banco de dados de Armazenamento de Sistema do IBM Security Directory Integrator. Por padrão, isso é configurado para 'Após cada operação do banco de dados' e, portanto, ocorre durante a fase de leitura.

No entanto, se você desejar garantir que uma mudança tenha sido transferida com êxito antes de confirmar a captura instantânea, configure a lista suspensa como 'No fim do ciclo do AL', de forma que ela seja efetuada depois que o destino de Saída tiver sido atualizado.

Para que o Mecanismo Delta execute esse trabalho, ele precisa de um conjunto de capturas instantâneas da linha de base. Você cria isso executando sua tarefa ETL pela primeira vez depois que o Delta foi ativado. Após a conclusão, você perceberá que o pop-up relata duas vezes, pois estão ocorrendo muitas gravações. Isso acontece porque o IBM Security Directory Integrator também conta as capturas instantâneas que estão sendo gravadas no Armazenamento do Sistema, de forma que você obtenha duas gravações para cada entrada processada.

Tente executar o EasyETL AssemblyLine novamente e você verá que nenhuma entrada foi gravada desta vez. O Mecanismo Delta detectou que nenhum registro de entrada foi alterado e os ignorou.

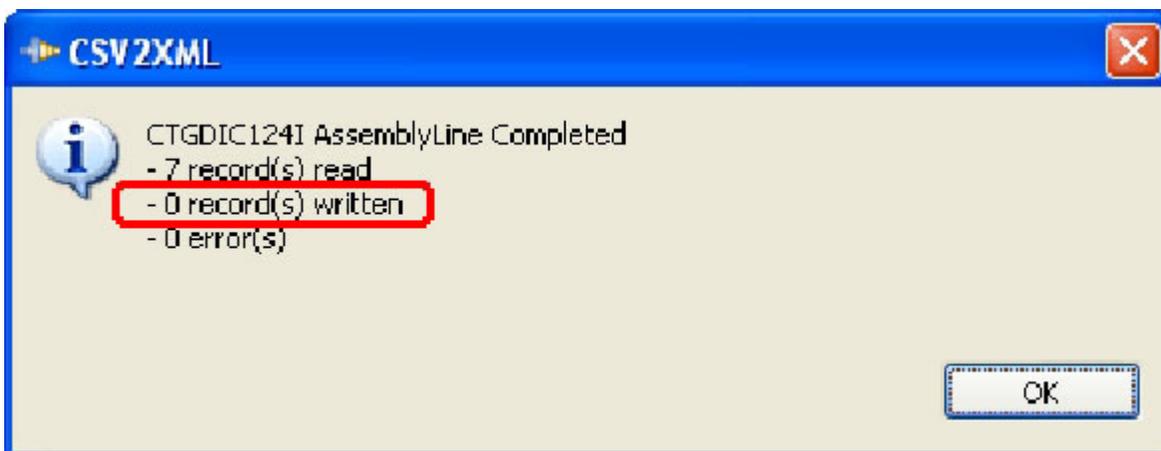


Figura 104. Todas as Entradas Inalteradas e Ignoradas

Como teste final, abra o arquivo CSV de entrada e altera qualquer um dos valores de campo – com exceção de 'Last'⁴⁷. Salve a mudança e execute novamente sua tarefa ETL e você verá que apenas as entradas modificadas são processadas.

Configurando o destino de saída para atualizações

A configuração atual funciona corretamente para saída para um arquivo. No entanto, se você estava direcionando essas mudanças para um diretório, um RDBMS ou um armazenamento de dados similar, convém incluir novos dados e atualizar registros existentes. Para sua tarefa do EasyETL executar isso, primeiro você deve selecionar que Atributo de Saída usar como os critérios para localizar o registro a ser modificado.

Isso é feito ao clicar com o botão direito do mouse no Atributo de Saída desejado e selecionar a opção **Usar como critérios de link**.

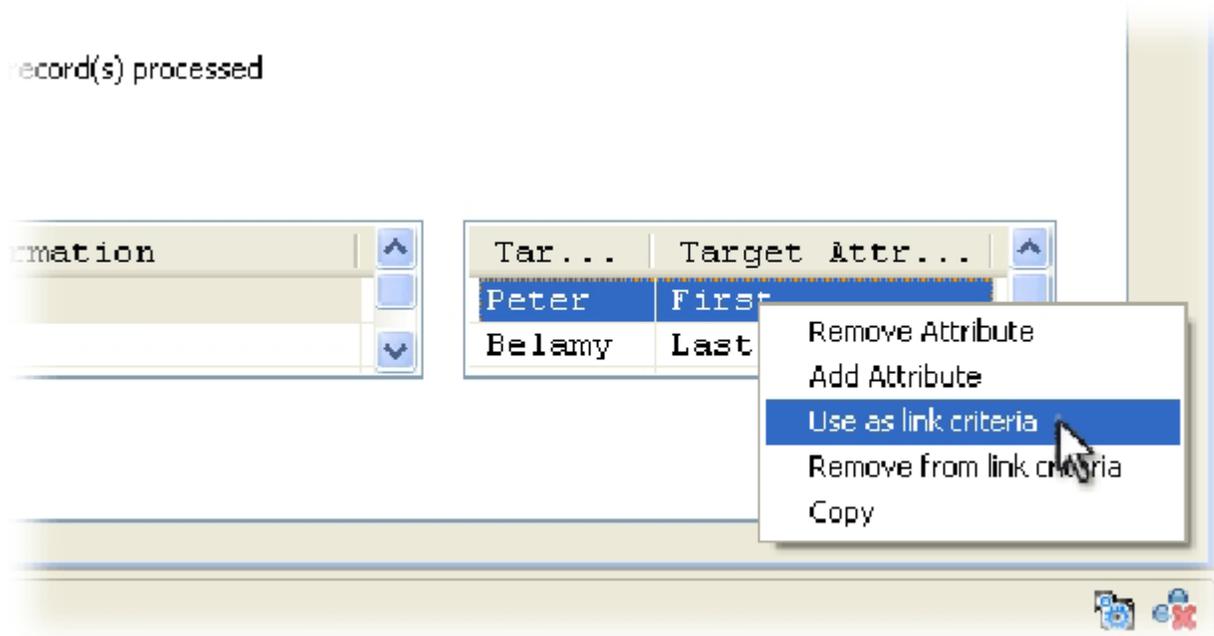


Figura 105. Selecionando os Critérios de Link

Agora, quando o Conector de Saída grava no destino, primeiro ele procura um registro usando o atributo Critérios de Link especificado. Se nenhuma correspondência for localizada, uma nova entrada será incluída. Se a correspondência obtiver êxito, esse registro será atualizado.

Isso é tão simples que: sua tarefa ETL agora foi configurada para fornecer sincronização contínua entre sua origem de entrada e o destino de saída.

Recursos da linha de comandos para executar e planejar a tarefa ETL

Depois que o ETL AssemblyLine estiver pronto para implementação, você poderá clicar com o botão direito do mouse no Projeto do Navegador e escolher a opção

47. Como este é o atributo usado para identificar capturas instantâneas, qualquer mudança em seu valor para uma entrada fará com que ele apareça como um *novo* registro para o Mecanismo Delta.

Criar arquivos necessários...

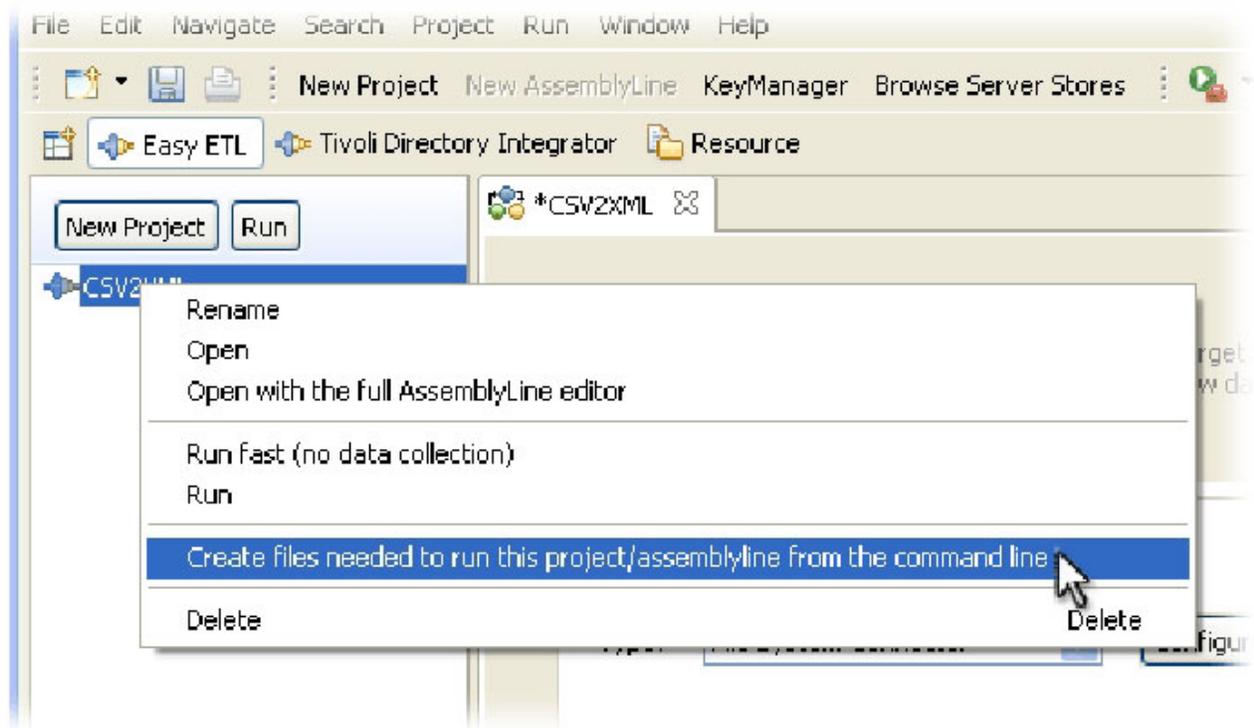


Figura 106. Criando recursos da linha de comandos para executar a tarefa ETL

O diálogo Exportar Arquivos é aberto, no qual é possível gravar script/arquivo em lote. Observe que será atribuído o mesmo nome do Projeto. Portanto, no caso do exercício deste tutorial ser executado em Windows, ele será chamado 'CSV2XML.bat'. A execução do Projeto do EasyETL a partir da linha de comandos proporciona o máximo desempenho para a sua solução.

Você também obterá um arquivo XML criado no mesmo local. Ele é chamado de arquivo *Config* do IBM Security Directory Integrator e contém os detalhes do seu EasyETL AssemblyLine que o Servidor IBM Security Directory Integrator precisa para executá-lo. Se abrir o script gerado em um editor de texto, você verá o comentário de linha única necessário para iniciar um Servidor IBM Security Directory Integrator, aponte-o em um Config e, em seguida, especifique o AssemblyLine a ser executado. Tudo o que você precisa fazer agora é configurar uma tarefa planejada ou uma tarefa cron para invocar periodicamente esse script e seu serviço de sincronização/migração será realizado.

Opções adicionais

ETL de Alta Velocidade

Embora o Data Collector seja uma ferramenta poderosa, o ETL AssemblyLine é executado mais lentamente devido à coleta de dados e à apresentação na tela. Se, em vez disso, você desejar que seu EasyETL AL processe o mais rápido possível, selecione o Projeto e pressione o botão Executar na parte superior do Navegador ou clique com o botão direito do mouse no Projeto e selecione a opção **Executar Rápido...**

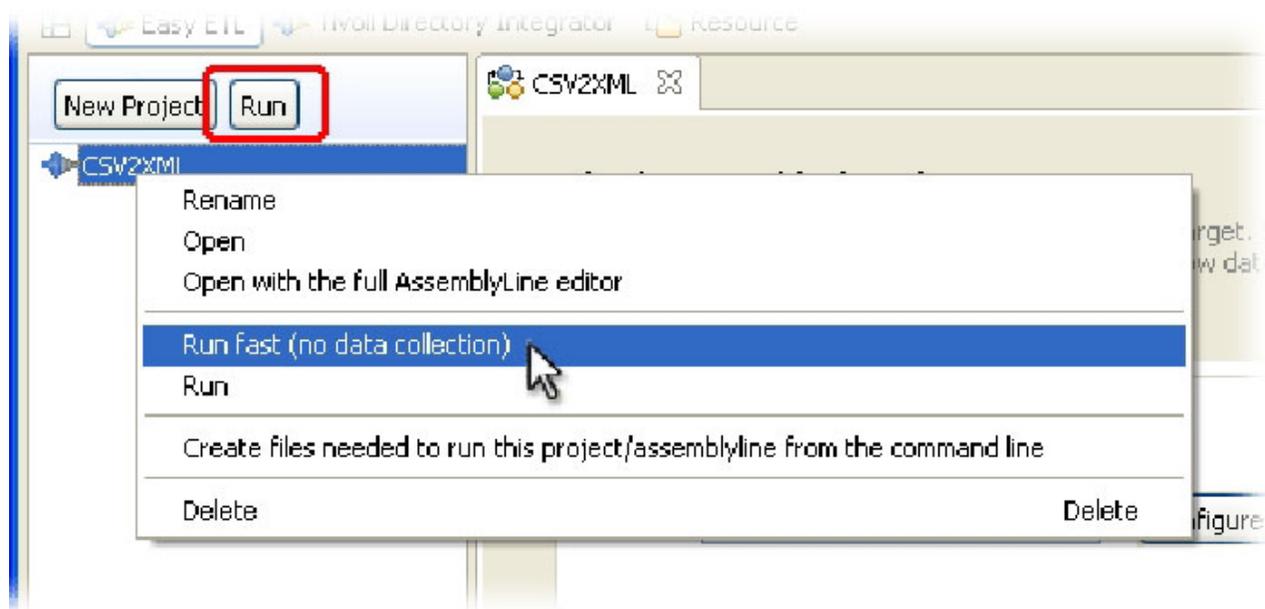


Figura 107. Executando a Tarefa ETL em Velocidade Máxima

Qualquer uma das opções abrirá uma exibição de console, em que as mensagens de log do seu AssemblyLine aparecerá à medida que o AL executar na velocidade máxima.

Observe que a opção **Executar** no menu de contexto Projeto executa a tarefa ETL com coleta de dados.

Filtrando o conjunto de dados de entrada

Outro recurso poderoso é a capacidade de controlar o conteúdo de seu conjunto de dados de Entrada. Essa opção está disponível sempre que sua origem de Entrada é um banco de dados ou diretório.

Por exemplo, selecione o 'Conector LDAP' para entrada e veja o diálogo de configuração deste componente. Ao lado do parâmetro **Filtro de Procura**, há um botão com três pontos (...). Clicar nesse botão abre o editor de Critérios de Link em que você pode definir as regras de procura que serão aplicadas para construir o conjunto de resultados para esse Conector ler.

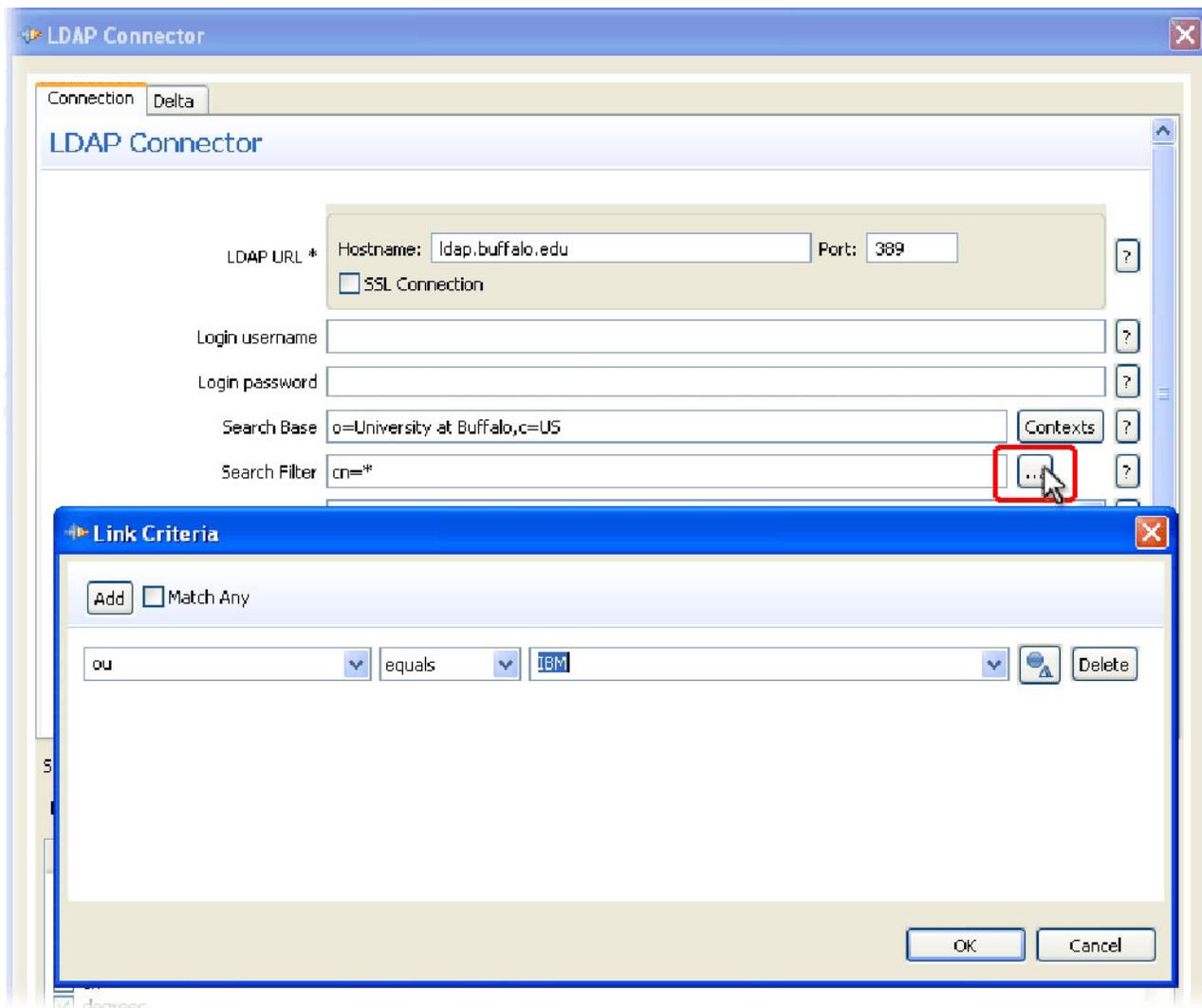


Figura 108. Definindo Critérios de Link para um Conector de Entrada

Esse mesmo recurso está disponível para o Banco de Dados e os Conectores JDBC, em que você localizará o botão (...) ao lado do parâmetro **Selecionar**.

Embora você mesmo possa inserir a sintaxe da procura de LDAP diretamente no parâmetro de procura, será necessário saber a sintaxe para os filtros de procura LDAP ou para instruções Select de JDBC. Normalmente, é mais simples expressar a seleção desejada ao usar os Critérios de Link e permitir que o Conector manipule a sintaxe subjacente.

Colocando o EasyETL AssemblyLine no próximo nível

Abrir o Projeto do ETL no editor completo do IBM Security Directory Integrator AssemblyLine permite incluir criação de log e auditoria customizada, manipulação de erros, lógica de failover, reconexão automática, aumento de dados (junções) e muito mais para sua solução de migração ou sincronização. Você faz isso clicando com o botão direito do mouse em um Projeto e escolhendo a opção **Abrir com o Editor**

AssemblyLine Completo. Você ainda continuará trabalhando no EasyETL Workbench, mas conseguirá obter a funcionalidade adicional disponível para o seu AssemblyLine.

Se esta for a sua preferência e você estiver pronto para tomar a iniciativa, alterne para a perspectiva do Security Directory Integrator (**Windows > Abrir Perspectiva > Security Directory Integrator**) e comece a trabalhar no IBM Security Directory Integrator Workbench integral. Ainda melhor - agora que você é um mestre em EasyETL, volte para a seção 1 e comece a tirar o máximo proveito do IBM Security Directory Integrator.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos neste documento em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não lhe garante direito algum sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Para consultas sobre licenças a respeito de informações do conjunto de caracteres de byte duplo (DBCS), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie consultas, por escrito, para:

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

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local:

A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO.

Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a Web sites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses Web sites. Os materiais contidos nesses Web sites não fazem parte dos materiais desse produto IBM e a utilização desses Web sites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Os resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intentos futuros da IBM estão sujeitas a mudanças ou retirada sem aviso e representam apenas metas e objetivos.

Todos os preços IBM mostrados são preços de varejo sugeridos pela IBM, são atuais e estão sujeitos a alterações sem prévio aviso. Os preços do revendedor podem variar.

Estas informações foram projetadas apenas com o propósito de planejamento. As informações aqui contidas estão sujeitas a alterações antes que os produtos descritos estejam disponíveis.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas. O Cliente pode copiar, modificar e distribuir estes programas de amostra de qualquer maneira sem pagamento à IBM com objetivos de desenvolvimento, uso, marketing ou distribuição de programas de aplicativos em conformidade com interfaces de programação de aplicativos da IBM.

Cada cópia ou parte destes programas de amostra ou qualquer trabalho derivado deve incluir um aviso de copyright com os dizeres:

© (nome da empresa) (ano). Partes deste código são derivadas dos Programas de Amostra da IBM Corp. © Copyright IBM Corp. _insira o ano ou anos_. Todos os direitos reservados.

Se estas informações estiverem sendo exibidas em cópia eletrônica, as fotografias e ilustrações coloridas podem não aparecer.

Marcas Registradas

IBM, o logotipo IBM e ibm.com são marcas ou marcas registradas da International Business Machines Corp., registradas em várias jurisdições no mundo todo. Outros nomes de serviços e produtos podem ser marcas registradas da IBM ou outras empresas. Uma lista atual de marcas registradas da IBM está disponível na web em "Copyright and trademark information" em www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript e todas as marcas registradas baseadas em Adobe são marcas ou marcas registradas da Adobe Systems Incorporated nos Estados Unidos e/ou em outros países.

IT Infrastructure Library é uma marca registrada da Central Computer and Telecommunications Agency, que agora faz parte do Office of Government Commerce.

Intel, o logotipo Intel, Intel Inside, o logotipo Intel Inside, Intel Centrino, o logotipo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium são marcas ou marcas registradas da Intel Corporation ou suas subsidiárias nos Estados Unidos e outros países.

Linux é uma marca registrada da Linus Torvalds nos Estados Unidos e/ou em outros países.

Microsoft, Windows, Windows NT e o logotipo Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

ITIL é uma marca registrada, e uma marca registrada da comunidade do departamento de comércio do governo, e está registrada no Departamento de Marcas e Patentes dos Estados Unidos.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.



Java e todas as marcas registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou de suas afiliadas.

Cell Broadband Engine é uma marca registrada da Sony Computer Entertainment, Inc. nos Estados Unidos e/ou em outros países e é usada sob licença.

Linear Tape-Open, LTO, o logotipo LTO, Ultrium e o logotipo Ultrium são marcas registradas da HP, IBM Corp. e Quantum nos Estados Unidos e em outros países.

Índice Remissivo

A

acessibilidade ix
acionar 71
add 8
AMC 71
anexador 83
arquivos do tutorial 1
AssemblyLine 6, 15, 30
Atributo 5

C

commandline 71
Comportamento Nulo 32
conectividade 83
conexão perdida 83
Conn 6
console 72
corresponder 66
Criação de Log 83
Critérios de Link 66

D

Dados 5
dados ausentes 32
Dados de consulta 48
Depuração 40
depurador 87
desempenho 86
determinação de problema ix
disponibilidade 81, 84

E

educação ix
Entrada 5
erro de inicialização 83
erros de Execução 67
escala 86
Evento 71
Eventos 71
exceções 84
Executar 30

F

fluxo de dados 6
fortalecendo 81

H

herança 63

I

IBM
Suporte de Software ix

IBM (*continuação*)

Support Assistant ix
iniciando soluções bem básicas 3
Introdução 8
iterar 8

J

JavaScript 1
junção 48

L

legibilidade 81
log4j 83
lookup 8

M

Manipulação de eventos 73
manutenção 81
Mapa de Atributos 6
Mapa de Entrada 6
Mapa de Saída 6
Mapeamento de Atributo 15
modo de Consulta 66
Modo de Consulta 59

P

passo a passo 40
planejador 72
planejando 72
procurar 66
projetando soluções 1, 3
projetos do IBM Security Directory
Integrator 1

R

resolução de problemas ix
responder 71
reutilização 81

S

saída do Log 30
segurança 81
Servidor da Web 73
soluções usando o IBM Security Directory
Integrator 3

T

TINA 73
tolerância a falhas 81
treinamento ix

V

Valor 5
valores Nulos 32



Impresso no Brasil

G517-9703-03

