software AG

webMethods Event Processing Help

Version 9.9

October 2015

WEBMETHODS

This document applies to webMethods Event Processing Version 9.9 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

About this Guide

This document gives you an overview of webMethods Business Events, which is Software AG's infrastructure for managing simple event-based interactions and more complex event analysis for pattern matching in real-time.

webMethods Business Events offers the following key features and functionality:

- It is a solution for creating, processing, and monitoring events.

- It provides the infrastructure to rapidly build and adapt event-driven applications.

- It improves an organization's ability to comprehend the current state of the physical world and business environment and react rapidly to changes.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service* .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.

Convention	Description
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.

- Access articles, code samples, demos, and tutorials.

- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.

- Link to external websites that discuss open standards and web technology.

1   Introduction to Event Type Development

Overview

Event types are XML schemas that define the structure of an event that components can publish or consume. An event type describes the event instances that a given input stream or output stream will carry.

An input stream identifies a stream of events from a specified topic on the event bus. An input stream has an associated event type, which describes the structure of the events that the input stream carries. An input stream can carry only one type of event.

You create event types using the Event Type Editor. If you want to reuse an event type that has already been defined, you can add it to your project from the Event Type Store.

If your organization is using CentraSite for SOA governance, you can add event types to, and retrieve event types from the CentraSite registry. If your organization publishes its event types to CentraSite, you can search the CentraSite registry for the event type you need.

The Event Type Editor

The Event Type Editor is an editor specially designed to create schemas for EDA applications. The schemas created by the Event Type Editor conform to the W3C XML Schema (XSD) specification, but they only use a subset of the available XSD features. This means that the Event Type Editor is specially tailored to allow you to create your event schemas simply and quickly, by limiting the functionality to just the XSD features that are required.

The schemas created by the Event Type Editor have the following structure:

- A standard XML header, as required by XSD. This is created automatically by the editor.

- A global element that contains all of the user-defined nodes. This is created automatically by the editor. This global element will be the payload of event instances that are processed in the EDA environment.

- The nodes you define. These can be field nodes, composite nodes or references to structures in other schemas. Field nodes are XSD simple type elements with a cardinality. Composite nodes are containers for field nodes and other composite nodes. References are pointers to top-level nodes in other schemas. For more information about elements types, see "Overview of Available Event Type Elements" on page 20.

Usage of the editor is simple: a palette contains the available node types, and you can drag and drop icons from the palette to the editor canvas to construct the required event type structure. After you have created any given node, use the **Properties** view in order to provide the required properties for the node type.

When you create a node that is a reference to a node in another schema, the editor assists you by presenting a list of schemas in predefined locations, and allows you to examine the structure of the referenced schema in order to find the node that you want to reference.

The Event Type Lookup Path

The event type lookup path is a list of paths visible to the compiler when building the project. The event type lookup path contains all of the following locations:

■ The current event type project.

■ Other workbench projects to which you explicitly declare a reference. You can add project dependencies by selecting other workbench projects to add to the lookup path for this project. To do so, use the **Project > Properties > Project References** preference page.

■ The local copy of the predefined event types. When you install webMethods Business Events, the predefined event types are copied to the folder common \PredefinedEventTypes below the installation directory.

The predefined event types in this folder are a local copy of the predefined types that are installed with the Event Type Store. If you change the contents of the Event Type Store, these changes are not reflected in the local copy of the predefined types. Similarly, if you change the local copy of the predefined types, these changes are not reflected in the Event Type Store.

Use of Relative Paths in the Event Type Definition

The use of the lookup path means that filenames used in the event type definition can be specified using relative paths rather than absolute paths. Consider for example the following import statement that is generated automatically by the Event Type Editor in each event type definition:

```
<xsd:import namespace="http://namespaces.softwareag.com/EDA/Event"
        schemaLocation="Event/Envelope.xsd"/>
```

Here, the Event Type Editor correctly treats Event/Envelope.xsd as a relative path within the local copy of the predefined types.

If you open an event type in a schema editor other than the Event Type Editor, the editor will be unable to resolve this path, since it is unaware of the event type lookup path. This can happen, for example, if you validate a project (by clicking **Validate** in the project's context menu in the Project Explorer view); in this case, Eclipse uses its own schema checking to compile the event type and will generate a warning that the relative path cannot be resolved. You can ignore all such validation messages.

2   Configuring the Event Type Editor Environment

Eclipse Preferences

You can tailor various aspects of your working environment by modifying program settings in the preferences page. The preferences page is available under **Window > Preferences**. The preferences for the Event Type Editor are available in the preferences page under **Software AG > Events Development**.

The following table summarizes the settings that you can modify.

Setting	Meaning
Undo history size	The maximum number of undo operations that you can perform in the Event Type Editor. The default value is set to `200`.
Console buffer size (characters)	The buffer size of the Event Bus Console. The default value is set to `10000000`.
NERV configuration file	The location of the Event Routing properties file that contains information about the Message Service component, the JNDI provider, and your Event Type Store location. Your Event Bus Console uses Event Routing to connect to the Event Bus and receive events. The default value is set to *Software AG_directory*\common\conf\com.softwareag.eda.nerv.properties. **Important:** During the creation of an Event Bus Console run configuration, if you have specified values for the **NERV Provider URL** and the **Event Type Store Location**, these values always take precedence over the values specified in the Event Routing properties file.

Working with Integration Server

The Integration Server is a webMethods product suite component. One of its features is to maintain a store of document types.

The Event Type Editor contains a feature that allows you to access the Integration Server store, read a document type definition and create a new event type based on the document type definition. For more information, see "Creating Event Types" on page 17.

To be able to use this feature in the Event Type Editor, the following prerequisites must be fulfilled:

■   The Integration Server must be installed and running.

To install the Integration Server, run the Software AG Installer and select the option **Designer>Integration Server> Server**.

To start the Integration Server after the installation, use the appropriate entry from the Windows Start menu.

■   The **Service Development** perspective must be installed.

To install this perspective, run the Software AG Installer and select the option **Designer > Services > Service Development**.

When you install the Integration Server, a connection profile is automatically created in the Software AG Designer. This means that whenever you start the **Service Development** perspective, a connection is automatically made from the Designer to the Integration Server.

Restrictions for Working with Integration Server Document Types

■   No 1:1 correspondence exists between document type definitions in the Integration Server store and event types used in the Event Type Editor. If a document type in the Integration Server contains a structure that is not available in the Event Type Editor, the creation of an event type from the document type is not possible. For example, Integration Server document types that contain nodes of `String Table`, `Object`, and `Object List` type cannot be converted to event types.

■   If you create a new event type in Software AG Designer from a selected Integration Server document type, the generated XML schema may not have a target namespace, depending on the selected Integration Server document type. This causes a warning message to appear in the Problems view within the Software AG Designer. If your application scenario does not require transforming documents that are valid instances of the initial document type into XML events that are valid instances of the event type's XML schema, you can change the target namespace of the event type schema by adding the target namespace, for example using a text editor. Otherwise, you can ignore the warning message.

You can also create a new Integration Server document type from the event type schema with the modified target namespace.

- If you create a new event type in the Software AG Designer starting with an Integration Server document type, additional files may be created. This occurs when the original document contains references to other Integration Server document types. By default, the additional files have the new event type name as prefix.

3 Working with Event Types

Overview

You can create event types quickly and simply using the Event Type Editor. The editor uses standard drag-and-drop technology that allows you to construct an event type by selecting standard schema components from a palette and placing them at the location of your choice in the event type schema.

The editor is available as a view in the **Events Development** perspective in the Software AG Designer.

Starting the Event Type Editor

To start the editor, proceed as follows:

To start the Event Type Editor

1. Start the Software AG Designer.

2. From the **Window** menu of the Software AG Designer, select **Open Perspective > Other**.

3. Select the **Events Development** perspective.

4. Open the **Event Type Editor** view.

Creating Event Type Projects

When you work with event types, the event types are grouped in an event type project. Each event type you create must be assigned to a project.

You can create any number of event type projects, and each project can contain many event types. Typically, a project is used as a container for a set of logically related event types.

To create an event type project

1. Open the **Project Explorer** view, if it not already open.

2. In the context menu of the **Project Explorer**, click **New > Event Type Project**.

3. In the dialog, enter a name for the new project, then click **Finish**.

The new project is now visible in the **Project Explorer** view.

The folder Event Types is created automatically under the project root node. This is the folder where you will store your event types for this project.

You can create further levels of subfolders under the Event Types folder by using the option **New > Folder** from the context menu of the parent folder.

Creating Event Types

After you have created an event type project, you can create an event type in the project as follows:

To create an event type in an existing project

1. In the **Project Explorer** view, select the project where you wish to create the event type.

2. In the context menu, click **New > Event Type**.

3. In the dialog, the name of the project is displayed. You can change this if you wish by selecting a different project name from the drop-down list.

4. In the field **Folder**, you can optionally enter the name of a folder where the new event type will be stored.

   If you enter a name, the event type will be stored in the given folder under the Event Types folder.

   If you do not enter a name, the event type will be stored directly in the Event Types folder.

5. In the field **Name**, enter the name of the event type.

6. If you have a connection profile for the Integration Server, the dialog additionally contains the option **Use an existing IS document type** that allows you to create an event type by importing a document type from the Integration Server. If you select this option, you can browse through the document types in the Integration Server store and select your required document type.

   For more information, see "Working with Integration Server " on page 13.

Note:	Please note that any attribute declarations existing in the IS document type will be removed when the Event Type is created.

7. Click **Finish**.

The new event type is now visible in the **Project Explorer** view.

The initial contents of the event type are displayed automatically in the Event Type Editor. Instructions on how to add your required structure to the event type using the Event Type Editor are provided in the following sections.

Opening an Existing Event Type

If you have already created an event type and you now wish to view or modify it, proceed as follows:

To open an existing event type

1. In Designer, open the **Project Explorer** view, if it is not already open.

2. Navigate to the required event type.

3. In the context menu, click **Open With > Event Type Editor**.

   This opens the event type in the Event Type Editor.

Renaming Event Types

You can rename an existing event type using the **Events Development** perspective in Designer.

When you use the rename feature to rename an event type, this not only renames the file but also where necessary adapts the contents of the file. This is because the contents of files of these types include a reference to the file name.

For this reason, if you wish to rename a file, use only the Rename feature provided rather than using the file renaming facility of the host operating system.

To rename an event type

1. In Designer, in the Project Explorer view, select the event type you want to rename.

2. Right-click and select **Rename**.

3. Enter the new name for the event type, keeping the `.xsd` extension, and click **OK**.

Saving Event Types

When you have made modifications to your event type, save it to disk by any of the standard Eclipse methods, such as using the key combination `Control-S`.

4   Working with Event Type Elements

Overview of Available Event Type Elements

The Event Type Editor provides several node types, and each node type has a set of properties associated with it. The available node types and their properties are shown in the following table:

Node Type	Description	Properties of the node type
Field	A node that contains an XSD simple type definition.	▪ Name ▪ Cardinality ▪ Type: the data type of the node. ▪ Filterable ▪ Documentation
Composite	A container for a set of nodes. The nodes can be field nodes or other composite nodes.	▪ Name ▪ Cardinality ▪ Documentation Nodes of this type have no data type, since they are just containers for other nodes.
Reference to Component	A reference to a node in another schema, which can be an XSD element, XSD simple type or XSD complex type. **Note:** Recursive schema references are not allowed. For instance, a schema should not reference itself or reference another schema that refers the current one.	▪ Name ▪ Cardinality ▪ Referenced Component ▪ Referenced Item ▪ Documentation Additionally, for a reference to a simple type: ▪ Filterable

The properties of the nodes have the following meanings:

▪ **Name**

The display name of the node on the canvas.

- **Cardinality**

The possible number of occurrences of this node. The available values are:

- 0..1, 0..n, 1..1, 1..n.

- **Type**

The data type of the node.

The available data types correspond to the simple data types of the XML Schema definition, as described on the W3C page at http://www.w3.org/TR/xmlschema-2/#built-in-datatypes.

> **Note:** The `anyType` and `anySimpleType` types are not supported.

- **Referenced Component**

The name of the referenced schema.

- **Referenced Item**

The name of the node in the referenced schema.

- **Filterable**

This is a boolean value that indicates whether or not the node value in instances of the event type can be used as a filtering criterion to select the instance for further processing. A value of `true` indicates that the node value can be a filtering criterion.

- **Documentation**

This property allows you to provide a comment explaining the purpose of the node.

Naming Conventions for Event Type Elements

When you assign names to the tree nodes in an event type, the following naming rules apply:

- The node names of all field nodes, composite nodes and reference nodes that reference an XSD element must be unique.

- If a reference node points to an XSD element, then the name is not editable.

- If a reference node points to an XSD simple type or XSD complex type, you can edit the name, provided that the new name is unique.

- EDA applications can have naming conventions that disallow the use of certain node names in the event type. For example, continuous query applications use the node names as field names in Event Query Language (EQL) statements, but EQL does not allow field names to be EQL keywords or Java type names. Therefore, when you create an event type, ensure that the node names comply with all such restrictions of the applications that will use the event type.

Adding Event Type Elements

You can add elements to the event type by dragging and dropping objects from the palette onto the editing canvas of the Event Type Editor. Alternatively, each element on the canvas has a context menu that allows you to perform operations such as adding elements before or after the selected element, moving the current element before the previous element or after the next element, or renaming or deleting the current element.

You can use standard cut/copy/paste functionality for editing the event type. The functionality for cut/copy/paste is available in the context menu of any selected node, and you can use the standard key combinations (Control-C etc.) for these operations also.

When you use the paste functionality, the behavior is as follows:

This behavior...	... performs this action
pasting on a composite node that is expanded	pastes the new node as the last subnode of the composite node
pasting on a composite node that is collapsed	pastes the new node as a sibling before the composite node
pasting on any other node	pastes the new node as a sibling before the selected node

The following table lists the possibilities for adding a node to the existing set of nodes on the canvas:

To do this...	... do this
Insert a node onto an empty canvas. (This situation occurs when you have just created an event type and are adding the first node, or if you deleted all of the nodes from an existing event type.)	Drag an object from the palette to any position on the empty canvas, and drop the object.
Insert a node as a sibling (i.e. as a child node of the same parent node) before/after an existing node.	■ **Drag and drop:** Drag an object from the palette to the location where you want to insert the node. As you drag the object, a horizontal line indicates where the node will be inserted. When the

To do this...	... do this
	horizontal line is at the required position, drop the object.
	If you want to insert a new node at the end of the event type, drag the object from the palette to any position on the empty canvas, and drop the object.
	■ **Context menu**
	Select an existing node on the canvas, then choose **Insert** from the context menu. In the Insert menu, choose the required node type, then choose **insert as sibling before** or **insert as sibling after** as required.
Insert a node as a child node of an existing node. (This operation is only possible if the existing node is a composite node.)	■ **Drag and drop:** If you want to add a node as the last subnode of a composite node, drop the object from the palette onto the icon of the composite node. If you wish to add a child node at a location other than the last subnode, expand the hierarchy of the existing composite node, so that the required position for the new node is visible, then drag the object from the palette to the required position. As an alternative to opening the hierarchy of the existing composite node explicitly, you can drag the new object and hover over the existing composite node for a few moments. This causes the composite node to expand automatically to reveal the hierarchy. Now continue to drag the new object and drop it at the required position. ■ **Context menu** Select an existing node on the canvas, then choose **Insert** from the context menu. In the Insert menu, choose the

To do this...	... do this
	required node type and **insert as a child node**.

Modifying Properties of Event Type Elements

After you have added a node to the event type, you can view and modify the node's properties in the **Properties** view of the perspective.

At run-time, when events of the given event type are transmitted over the event bus, event consumers subscribing to the event bus to receive events can apply a filter, so that only events that match certain selection criteria are consumed. Such criteria can be, for example, the topic name (which must always be provided), properties in the event header such as the start and end timestamp, and filterable properties selected from the body of the event. The `Filterable` property of the node in the event type indicates which values from the event body are available for filtering.

To modify the properties of an event type element

1. With the node selected, open the **Properties** view.

2. For a field node or a composite node, enter the new value or values as required, then save your changes.

   For a reference node, continue with the following steps.

3. For the `Referenced Component` property, use the **Browse** button to list all schemas that are located in the event lookup path.

   The list can show both schemas and event types. Schemas that are valid and that can be selected are shown with an icon with a white background. Event types are shown with an icon with a colored background.

   The field labeled **Choose an Object that will be referenced** is a type-ahead search field that you can use to search for a schema that matches the text you enter. You can use wildcards at the beginning or at the end of the field. If, for example, you specify `obj*` as the name of the schema, the display with show all schemas in the lookup path that begin with the letters `obj`.

   When you have chosen a schema, click **OK**.

   > **Note:** Recursive schema references are not allowed. For instance, a schema should not reference itself or reference another schema that refers the current one.

4. After you specify a value for the `Referenced Component` property, the drop-down list for the **Referenced Item** property is automatically updated to contain the names of the available top-level nodes for the chosen referenced schema. The top-level nodes can be XSD elements, XSD simple types or XSD complex types. This is indicated in

the drop-down list by the notation `[Element]`, `[Simple Type]` or `[Complex Type]` after the node name.

Select the required node from the drop-down list.

5. Supply values for the node name and cardinality as required.

6. Save your changes.

Viewing the Hierarchical Structure of Event Type Elements

The event type definition can contain hierarchical structures. These structures can be one of the following types:

■ composite nodes

■ references to objects in component schemas (complex types or elements)

These structures can be displayed in the tree view of the Event Type Editor as expanded or collapsed nodes. When a node is expanded, all of the child nodes are displayed in the tree view. When a node is collapsed, only the top-level composite node or reference node is displayed.

When you expand a composite node, you can edit any of the displayed child nodes using any of the editing possibilities described in the previous sections.

When you expand a reference to a complex type or an element in a component schema, the editor displays the child nodes as read-only nodes. These nodes belong to the component schema, so you can include them in your event type but you cannot edit them in any way. The display shows the name of each child node, followed by the data type of the child node in brackets.

When you click on any of the child nodes belonging to a reference node in the tree view, the properties of the child node are displayed in the **Properties** view. The properties are read-only.

Repositioning Event Type Elements

You can move any existing node in the event type from its current position on the canvas to another position. You can do this as follows:

■ Using drag and drop

■ Using the move up, move down, move left or move right commands in the node's context menu.

Drag and drop from one position on the canvas to another works in the same way as drag and drop from the palette to the canvas. For more information, see .

When you use move up/down/left/right from the context menu of a selected node, you move the node one position in the given direction, as described in the following table:

Direction	Description
Move up	This moves the selected node to before the previous sibling node.
	If there is no previous sibling node, the menu item is disabled.
Move down	This moves the selected node to after the following sibling node.
	If there is no following sibling node, the menu item is disabled.
Move left	This command is only possible if the selected node is a child node of a composite node, otherwise the menu item is disabled.
	This moves the selected node one level higher in the tree structure. The node becomes the immediate preceding sibling node of the composite node that was originally its parent node.
Move right	This command is only possible if there is an immediate following composite node, otherwise the menu item is disabled.
	This moves the selected node one level deeper in the tree structure. The node becomes a child node of the immediate following composite node.

Renaming Event Type Elements

You can rename any node by selecting it, then choosing **Rename** from the context menu. Alternatively, you can double-click the node name, then overwrite the existing name.

Deleting Event Type Elements

If you want to delete nodes from the event type, you can do so as described in the following table.

To do this ...	... do this
Delete a field node.	Select the node and choose **Delete** from the context menu.
	Alternatively, select the node and type `Control-X`.
Delete a composite node.	Select the node and choose **Delete** from the context menu.
	Alternatively, select the node and type `Control-X`.
	This deletes the composite node and its hierarchy.
Delete a reference to a component.	Select the node and choose **Delete** from the context menu.
	Alternatively, select the node and type `Control-X`.
	This deletes just the reference, not the node to which the reference points.

5   Publishing Event Types to CentraSite

Overview

You can publish an event type schema to CentraSite. This means that the event type schema is copied to the CentraSite Repository, and an asset of the type XML Schema for the schema is created in the CentraSite Registry.

Prerequisites for Connecting to CentraSite

Software AG Designer can connect to CentraSite if you selected the optional CentraSite plugins during the Software AG Designer installation. The plugins are installed if you select **Designer> SOA Governance >CentraSite***n.n*in the Software AG Installer, where *n.n* is the CentraSite version number.

If you did not select the optional CentraSite plugins during the Software AG Designer installation, none of the CentraSite-related features are available.

Configuring the Connection to CentraSite

Before you can publish Complex Event Processing components in CentraSite, you must define the required CentraSite connection parameters, and set the configured connection to be the active configuration.

Defining the CentraSite Connection Parameters

To define the CentraSite connection parameters

1. In the Complex Event Processing perspective, select **Window > Preferences**.

2. Locate the node **CentraSite**. Expand this node and select the subnode **Connections**.

   This opens the **Connections** dialog, which shows the currently configured connections to CentraSite servers.

   You can create a new configuration by clicking **Add**, or modify an existing configuration by clicking **Edit**.

3. In the field **Name**, specify a name of your choice that will be used to identify the configuration.

   In the field **Host**, specify the URL of CentraSite's configuration file. For a local CentraSite installation, this is usually `http://localhost:53305/ INMConfiguration/INMConfiguration.xml`.

   In the field **User**, specify the name of a CentraSite user who has the required CentraSite permissions to register an asset in the CentraSite registry.

If you wish, you can supply the password of the CentraSite user, or you can omit the password at this time. If you supply the password, it is stored in the Eclipse secure storage and will be supplied to CentraSite every time CentraSite requires your user authorization. If you do not supply the password, CentraSite will prompt you for the password each time your user authorization is required.

To supply the password in the configuration, mark the checkbox **Save password (in the Eclipse secure storage)** and supply the password of the CentraSite user in the field **Password**.

4. Click **Test** to check that the CentraSite server can be accessed using the configuration parameters.

   Click **OK** to store the configuration.

You can define several such configurations to CentraSite. However, only one of the configurations can be active at any given time. To set a configuration to be the active configuration, proceed as follows:

Setting a CentraSite Configuration as Active

To set a configuration to be the active configuration

1. Open the **Connections** dialog, as described above.

2. Mark the checkbox of the configuration that you want to make the active configuration.

3. Click **OK**.

Removing a CentraSite Configuration

To remove a configuration from the list, proceed as follows:

To remove a configuration from the list of configurations

1. Open the **Connections** dialog, as described above.

2. If the configuration that you want to delete is the active configuration (i.e the configuration whose checkbox is marked), you must first select another configuration as the new active configuration. For details on how to do this, see .

3. Select the row containing the configuration that you want to delete. Make sure that the checkbox for the row is NOT selected.

4. Click **Remove**.

Note:	You cannot delete the currently active configuration. If the list contains only one configuration, this configuration becomes automatically the active one, and you cannot delete it.

Publishing Event Types to CentraSite

You can publish an event type in CentraSite.

To publish an event type in CentraSite

1. Locate the required event type in the Project Explorer.

2. From the context menu, choose **Publish to CentraSite**.

   This menu item is only available if you have installed the CentraSite plugins.

3. In the **Publish to CentraSite** dialog, select the name of the CentraSite organization where you want to publish the event type, then click **OK**.

   The list of available organizations includes only the organizations for which you have the CentraSite permissions required to register an asset.

4. The event type will now be published in the CentraSite registry.

   After the event type has been published, it is visible in CentraSite as an asset of type XML Schema.

If the event type file in the Software AG Designer environment is registered in a version control system, this information is added automatically as an attribute to the CentraSite asset. Each asset of type Event Type in CentraSite has an attribute `VCS Location`, that contains the URL of the event type's location in the source control system. This value is copied from the Eclipse property `Repository URL` for the event type file. If the event type file in Software AG Designer is not registered in a version control system, the value of the CentraSite attribute `VCS Location` is not set. This feature is currently supported for the following version control system clients:

- Subclipse (http://subclipse.tigris.org/)

- Subversive (http://www.eclipse.org/subversive/)

- Eclipse CVS (http://www.eclipse.org/subversive/)

Verifying Event Type Is Published

You can check whether the event type was correctly published in CentraSite.

To check that the event type was published

1. Open the CentraSite view **Registry Explorer**, if it is not already open.

   In this view, expand the **Organizations** node. This shows the CentraSite organization to which the user defined in the connection configuration belongs.

   Expand the node of the CentraSite organization.

2. Expand the node **XML Schema**. This is the location into which event types are published.

   If your event type is visible in the list, then the event type has been successfully published. If you do not see the event type in the list, you may need to refresh your **Registry Explorer** view.

Unpublishing an Event Type in CentraSite

You might wish to unpublish an event type that you previously published in CentraSite. When you unpublish an event type, you remove the registration of the event type from the CentraSite registry and repository, but you do not delete or modify any information in the CEP project.

To unpublish an event type in CentraSite

1. In the Project Explorer view, locate the event type that you want to unpublish in CentraSite.

2. From the context menu of the event type, select **Unpublish from CentraSite**.

   This menu item is only available if you have installed the CentraSite plugins.

   The event type will now be removed from the CentraSite registry and repository.

6   Viewing Events using the Event Bus Console

Overview

The Event Bus Console tool is a Java-based utility that enables you to subscribe to channels on the Event Bus. It displays a trace of the event stream for these channels to the Eclipse console, enabling you to monitor in real time the typical traffic on the Event Bus.

Some start parameters, such as the server name and port number of the event bus server, are mandatory. You must define a launch configuration type `Event Bus Console` that you can use to create a new run configuration to execute the utility.

Prerequisites

To monitor the Event Bus traffic in the console, you must ensure that:

- The Event Bus server is running.

- The Event Bus contains at least one active channel.

- The active channel contains events which can be displayed in the Event Bus Console.

Creating an Eclipse Run Configuration

To connect to the Event Bus using the Event Bus console tool, you must create a run configuration in Software AG Designer. The run configurations define the start parameters for connection to the Event Bus server and the channels which you want to monitor. You can define more than one configurations to monitor different channels on different Event Bus servers.

To create a run configuration

1. In the Events Development perspective in Software AG Designer, click the **Create New Event Bus Console** icon.

2. On the **Main** tab of the **Run Configurations** dialog, specify the Event Routing connection settings for the Event Bus Console.

   - **NERV Provider URL** - defines the transport layer which NERV uses to connect to the Event Bus. The default value is `nsp://`*<host_name>*`:9000`. You can specify any valid URL of webMethods Universal Messaging or webMethods Broker to be used as the Event Bus server for receiving and subscribing for events.

   - **Event Type Store Location** - contains the path to the Event Type Store location. The default value is: *Software AG_directory*`/common/EventTypeStore`. You can use the **Browse** button to select another directory.

3. Click **Connect**.

4. On the **Channels** tab, specify the channels to which you want to subscribe. You can select JNDI topics and/or event types using the respective check boxes. You can use the **Select All** and **Deselect All** buttons on the right side of the dialog to mark or unmark all topics and event types simultaneously.

5. On the **Message Format** tab, modify the format in which the console will display the received events.

Select this option...	To display...
Show channel name as prefix for each line on the console	The channel name as prefix for each event on the console.
Show event type name as prefix of each line on the console	The event type name as prefix for each event on the console.
Show only every hundredth event on the console	The full details of every 100th event. The other events appear in the console only as a dot.
Show JMS properties	All the filterable properties that are in the JMS header. The XML element names in the event and the label used for the elements in the console output are mapped as follows:

XML element name	Label used in the output
eda:Event/eda:Header/ eda:Type	$Event$Type
eda:Event/eda:Header/ eda:Start	$Event$Start
eda:Event/eda:Header/ eda:End	$Event$End
eda:Event/eda:Header/ eda:Kind	$Event$Kind
eda:Event/eda:Payload/ root/@att	$att

Select this option...	To display...	
	eda:Event/eda:Payload/ root/child	child
	eda:Event/eda:Payload/ root/child/@att	child$$att

6. Click **Apply** to store the configuration.

Running the Event Bus Console

You can run the Event Bus Console tool in the **Events Development** perspective of Software AG Designer.

To run the Event Bus Console

1. In the **Events Development** perspective in the Software AG Designer, click the **Create New Event Bus Console** icon.

2. In the **Run Configurations** dialog, select the required run configuration under **Event Bus Console**, and click **Run**.

A sub-view opens in the Software AG Designer Console view. It displays events arriving on the Event Bus.

> **Note:** You can have several run configurations active at the same time. Each run configuration opens in its own sub-view in the Console view. You can use the icons in the Console view to switch between sub-views. Use the **Close** button in the Console view to stop an active run configuration.

7   Working with the NERV Example Wizards

Overview

The Events Development perspective in Software AG Designer provides several wizards for creating example NERV component and emit bundles. We recommend that you use the Designer wizards to ensure that your custom NERV bundles have the correct directory structure. Later you can modify the content of the component.xml and emit.xml files created by the wizards and provide your custom configurations.

Creating a NERV Component Example

Use this wizard to create an example of a custom NERV component configuration bundle.

To create a NERV default component example

1. In Designer, click **File > New > Example**.

2. Expand the NERV node, select the NERV Default Component Example wizard, and click **Next**.

3. Select the component.nervDefaultJMS project and click **Next**.

4. On the **Component Configuration** dialog, specify your webMethods Universal Messaging or webMethods Broker configuration settings, and click **Finish**.

The example has been created and added to your workspace. You can see it in the **Project Explorer** view in the Events Development perspective.

You can customize the blueprint.xml and the component.xml files to change the NERV component configuration settings, and then deploy the new configuration bundle that will override the default NERV component definition. For more information about creating NERV component configuration bundles, see the PDF publication *Implementing Event-Driven Architecture with Software AG Products*.

Creating a NERV Emit Example

Use this wizard to create an example of a custom NERV emit configuration bundle.

To create a NERV emit example

1. In Designer, click **File > New > Example**.

2. Expand the NERV node, select the NERV Emit Example wizard, and click **Next**.

3. Select the emit.NERVEmitExample project and click **Next**.

4. On the **Emit Configuration** dialog, browse to the Event Type Store, select an event type to be used for the emit configuration, and click **Finish**.

The example has been created and added to your workspace. You can see it in the **Project Explorer** view in the Events Development perspective.

You can customize the blueprint.xml and the emit.xml files to change the NERV emit configuration settings, and then deploy the new emit configuration bundle that will override the default NERV emit configuration definition. For more information about creating NERV emit configuration bundles, see the PDF publication *Implementing Event-Driven Architecture with Software AG Products*.

Creating a NERV Emit Complex Example

Use this wizard to create a complex example of a custom NERV emit configuration bundle.

To create a NERV emit complex example

1. In Designer, click **File > New > Example**.

2. Expand the NERV node, select the NERV Emit Complex Example wizard, and click **Next**.

3. Select the emit.NERVEmitComplexExample project and click **Next**.

4. On the **Emit Configuration** dialog, browse to the Event Type Store, select an event type to be used for the emit configuration, and click **Finish**.

The example has been created and added to your workspace. You can see it in the **Project Explorer** view in the Events Development perspective.

Browse to the /OSGI-INF/blueprint/emit.xml file and open it in Designer. You see that when deployed, this emit configuration bundle:

■ Sends events to the JMS topic;

■ Saves copies of the events on the file system in the *<root_drive>*/MyLogsDirectory directory;

■ Sends copies to the specified e-mail address.

You can customize the blueprint.xml and the emit.xml files to change the NERV emit configuration settings, and then deploy the new emit configuration bundle that will override the default NERV emit configuration definition. For more information about creating NERV emit configuration bundles, see the PDF publication *Implementing Event-Driven Architecture with Software AG Products*.