



# ASP.NET Entry Points

- [Default Automatic Naming for ASP.NET Transactions](#)
  - [Customize the Automatic Naming Scheme](#)
    - [To modify automatic naming](#)
- [Identify Transactions Using URI Segments](#)
  - [To name transactions using all, first, or last URI segments](#)
  - [To use specific URI segments in transaction names](#)
- [Identify Transactions Using Headers, Cookies, and Other Parts of HTTP Requests](#)
  - [To use HTTP parameter values in transaction names](#)
  - [To use a header value in transaction names](#)
  - [To use a cookie value in transaction names](#)
  - [To use a session attribute value in transaction names](#)
  - [To use the request method in Transaction names](#)
  - [To use the request host in Transaction names](#)
  - [To use the request originating address in Transaction names](#)
- [Custom Match Rules for ASP.NET Transactions](#)
  - [To create an ASP.NET custom match rule](#)
  - [To split custom ASP.NET transactions](#)
- [Learn More](#)

AppDynamics automatically detects entry points for client requests to ASP.NET applications. If the request occurs on an [originating tier](#), the method or operation marks the beginning of a business transaction and defines the transaction name. In most cases, this type of entry point maps to a user request or action such as "Cart/Checkout". AppDynamics allows you to configure transaction naming based upon the ASP.NET request.

## Default Automatic Naming for ASP.NET Transactions

By default, the AppDynamics auto-detection naming scheme identifies all ASP.NET transactions using the first two segments of the URI.

For example, the following URI represents the checkout operation in an online store:

```
http://mydotnetsite.com/Cart/Checkout
```

AppDynamics automatically names the transaction:

```
/Cart/Checkout
```

For another example, the following URI represents a funds transfer operation in an online bank:

`http://webbank.mybank.com/Account/Transferfunds/NorthernCalifornia`

AppDynamics automatically names the transaction:

`/Account/Transferfunds`

## Customize the Automatic Naming Scheme

The AppDynamics auto-detected transaction names might not be optimal for your users. You can configure the naming scheme as follows:

- [Identify transactions using URI segments](#)
- [Identify transactions using headers, cookies, and other parts of HTTP requests](#)

### To modify automatic naming

1. Click **Configure > Instrumentation > Transaction Detection**.
2. Click the **.NET - Transaction Detection** tab.
3. From the **Select Application or Tier** list at the left, click either:
  - an application to configure transaction detection for all tiers in a business application.
  - a tier. At the tier level click **Use Custom Configuration for this Tier**. AppDynamics copies the application configuration to the tier level so that you can modify it for the tier.
4. If necessary, click **Enabled** under Transaction Monitoring and **Discover Transactions automatically for ASP.NET requests**.  
 You can configure naming with Discover Transactions automatically for ASP.NET requests disabled, but the agent doesn't discover ASP.NET transactions.
5. Click **Configure Naming** for the ASP.NET type in the in the Entry Points panel.
6. Change the naming scheme in the ASP.NET Transaction Naming Configuration window and click **Save**.

The following sections provide examples to help you decide how to configure the naming scheme.

## Identify Transactions Using URI Segments

AppDynamics offers the following options to automatically name ASP.NET transactions based upon the URI:

- [Use all, first, or last URI segments](#)

- [Use specific URI segments](#)

### To name transactions using all, first, or last URI segments

Consider the following URL that represents the checkout operation in an online store:

```
http://mydotnetsite.com/Web/Store/Checkout
```

The first two segments of the URI don't provide a significant name for the business transaction:

```
/Web/Store
```

Identify a more meaningful name using one of the following options:

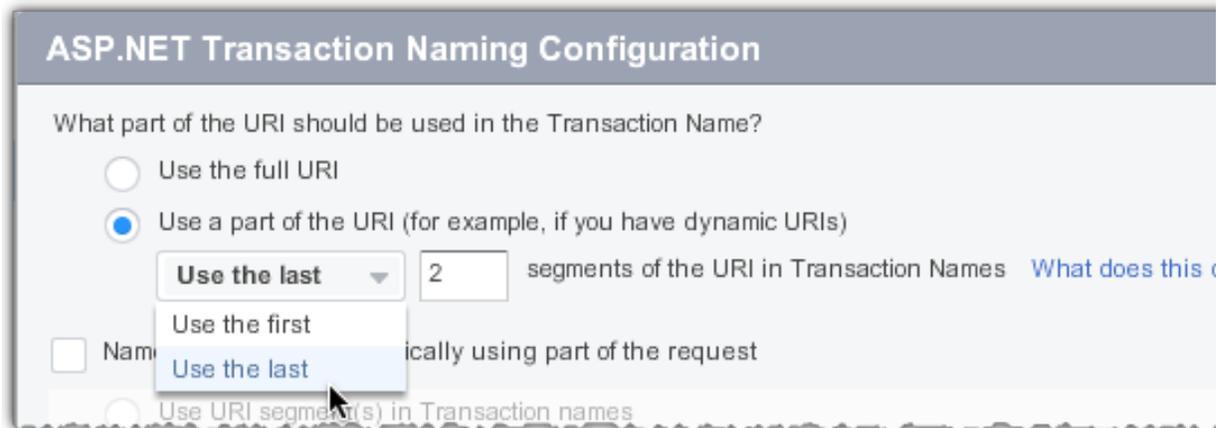
- Click **Use the full URI** to identify the transaction by all URI segments. For example:

```
/Web/Store/Checkout
```

- Click **Use the first** or **Use the last**  $n$  segments to use two contiguous segments at the beginning or end of the URI, where  $n$  is the number of segments.

For example, to identify the checkout transaction using the last two segments of the URI:

```
/Store/Checkout
```



- If you need more flexibility, such as using non-contiguous segments in the name, click **Name Transactions dynamically using part of the requests** and specify the segments with the **Use URI segments in Transaction names** option.

### To use specific URI segments in transaction names

You can choose specific URI segments to use in the transaction name. This enables you to skip URI segments or use non-contiguous segments in the naming scheme.

1. Click **Use a part of the URI**.
2. Enter the number of first or last segments to use.
3. Click **Name Transactions dynamically using part of the request**.
4. Click **Use URI segment(s) in Transaction names**.
5. Enter the segment numbers separated by commas.

For example the following URL represents the checkout transaction requested by a customer with ID 1234:

```
http://mydotnetsite.com/Store/cust1234/Checkout
```

The checkout transaction is the same regardless of the customer, so it makes sense to name the transaction based upon the first and third segments of the URI.



The screenshot shows the 'ASP.NET Transaction Naming Configuration' dialog box. It contains the following elements:

- Title: ASP.NET Transaction Naming Configuration
- Question: What part of the URI should be used in the Transaction Name?
- Radio buttons:
  - Use the full URI
  - Use a part of the URI (for example, if you have dynamic URIs)
- Dropdown menu: Use the first
- Text input: 1
- Text: segments of the URI in Transaction Names
- Link: What does th
- Checked checkbox: Name Transactions dynamically using part of the request
- Radio button:  Use URI segment(s) in Transaction names
- Text input: 3
- Text: Segment Numbers
- Text: Enter a comma separated list of parameter numbers (e.g. 1,3)

AppDynamics names the transaction:

```
/Store/Checkout
```

## Identify Transactions Using Headers, Cookies, and Other Parts of HTTP Requests

You can also name ASP.NET transactions using parameters, headers, cookies, and other parts of HTTP requests.

To identify all your ASP.NET transactions using particular parts of the HTTP request, use the **Name Transactions dynamically using part of the request** option.

**i** Carefully consider your naming configuration choices. If you use a value such as the request originating address and you have many clients accessing your application, you may see the [All Other Traffic Business Transaction](#).

### To use HTTP parameter values in transaction names

1. Set the URI identification option.
2. Click **Use a parameter value in Transaction names** and enter the **Parameter Name**.

For example, consider the following URL:

```
http://mydotnetsite.com/Store/Inventory?category=electronics
```



The screenshot shows a configuration panel with a torn-paper edge. At the top, there is a checked checkbox labeled 'Name Transactions dynamically using part of the request'. Below it are two radio button options. The first is 'Use URI segment(s) in Transaction names' with a text input field for 'Segment Numbers' and a placeholder text 'Enter a comma separated list of parameter numbers (e.g. 1)'. The second is 'Use a parameter value in Transaction names', which is selected. Below this selected option is a text input field for 'Parameter Name' containing the value 'category'.

AppDynamics names the transaction to include the category parameter value:

```
/Store/Inventory.electronics
```

### To use a header value in transaction names

1. Set the URI identification option.
2. Click **Use header value in transaction names** and enter a **Header Name**.  
For example, consider a site that uses the custom header "Version", AppDynamics names transactions with the header value as follows:

```
/Store/Inventory.v2.5
```

### To use a cookie value in transaction names

1. Set the URI identification option.
2. Click **Use a cookie value in Transaction names** and enter the **Cookie Name**.  
For example, a website tracks a user's loyalty status in a cookie. Set the Cookie Name to "loyalty". AppDynamics names transactions for the loyalty cookie value:

```
/Store/Inventory.Status=Gold
```

### To use a session attribute value in transaction names

1. Set the URI identification option.
2. Click **Use a session attribute in Transaction names** and enter the **Session Attribute Key**.

For example, a website stores a customer's region in the session property. Set the Session Attribute name to "region". AppDynamics names transactions for the region session attribute value:

```
/Store/Inventory.NorthAmerica
```

### To use the request method in Transaction names

1. Set the URI identification option.
2. Click **Use the request method (GET/POST/PUT) in Transaction names**. AppDynamics names transactions for the request method. For example:

```
/Store/Inventory.GET
```

### To use the request host in Transaction names

1. Set the URI identification option.
2. Click **Use the request host in Transaction names**. AppDynamics names transactions for the ip address of the request host. For example:

```
/Store/Inventory.192.0.2.0
```

### To use the request originating address in Transaction names

1. Set the URI identification option.
2. Click **Use the request originating address in Transaction names**. AppDynamics names transactions for the ip address of the request client. For example:

```
/Store/Inventory.192.0.2.10
```

## Custom Match Rules for ASP.NET Transactions

Custom match rules provide greater flexibility for transaction naming. When you define a match rule, AppDynamics uses the rule name for the business transaction name.

For steps to access the **Custom Match Rules** pane, see [To create custom match rules for .NET entry points](#).

### To create an ASP.NET custom match rule

1. In the **Custom Match Rules** pane, click the plus symbol (+) to add an entry point.
2. Click **ASP.NET** in the dropdown list. Click **Next**.
3. Name the **New Business Transaction Match Rule**.
  - AppDynamics uses the rule **Name** to name the BT.
  - The Controller enables the rule by default. Disable it later if needed.
  - Set the **Priority** for the match rule. AppDynamics applies higher priority rules first.
4. Set one or more of the following match criteria. When AppDynamics detects a requests matching your specified criteria, it identifies the request using your custom name.

**Method:** Match on the HTTP request method, GET, POST, PUT or DELETE.

 With automatic discovery for ASP.NET transactions enabled, configuring the match on GET or POST causes the the agent to discover both GET and POST requests. If you only want either GET or POST requests for the transaction, consider the following options:

- Disable automatic discovery for ASP.NET transactions.
- Create an exclude rule for the method you don't want: GET or POST.

**URI:** Set the conditions to match for the URI.

- For rules on regular expressions for .NET, see [.NET Framework Regular Expressions](#).
- Optionally click the gear icon to set a NOT condition.
- You must set an URI match condition in order to use [transaction splitting](#).

**HTTP Parameter:** Match on HTTP parameter existence or a specific HTTP parameter value.

**Header:** Match on a specific HTTP header's (parameter's) existence or a specific HTTP header value.

**Hostname:** Match on the server host name. Optionally click the gear icon to set a NOT condition.

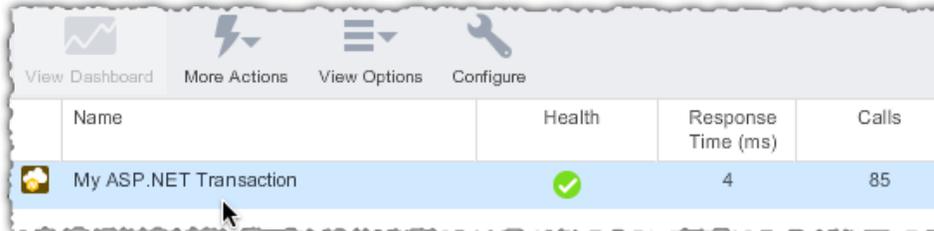
**Port:** Match on the server port number. Optionally click the gear icon to set a NOT condition.

**Class Name:** Match on the ASP.NET class name. Optionally click the gear icon to set a NOT condition.

**Cookie:** Match on cookie existence or a specific a specific cookie value.

5. Click **Save**.

The rule appears in the **Custom Match Rule** list. The business application or tier you customized displays a green check in the **Select Application or Tier** pane. After the agent receives the updated configuration, it discovers the new business transaction and displays it in the [Business Transactions List](#).



Name	Health	Response Time (ms)	Calls
My ASP.NET Transaction	✓	4	85

### To split custom ASP.NET transactions

AppDynamics lets you further refine ASP.NET custom transaction names using [transaction splitting](#). See [Transaction Splitting for Dynamic Discovery](#).

1. Create a custom match rule. To use transaction splitting, you must specify URI match criteria.
2. Click **Split Transactions Using Request Data**.
3. Click the splitting option to use.

The transaction splitting options work the same as the methods described in the previous sections:

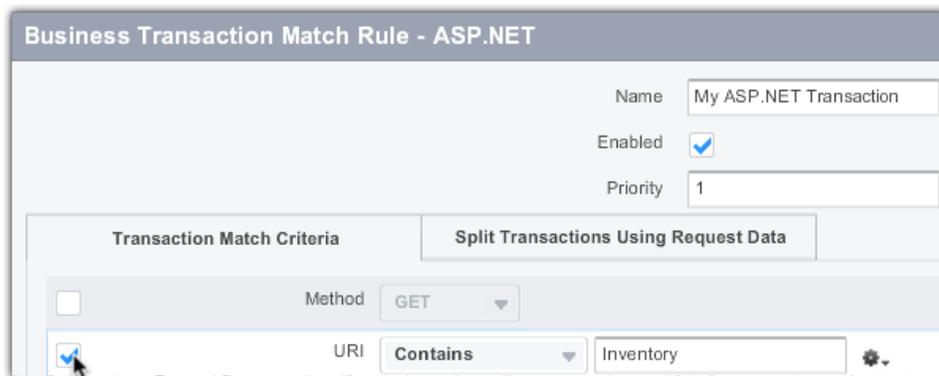
[Identify transactions using URI segments](#)

[Identify Transactions Using Headers, Cookies, and Other Parts of HTTP Requests](#)

For example, consider the following URL:

`http://mydotnetsite.com/Store/Inventory?category=electronics`

Configure the custom match rule to match on the "URI contains Inventory".



Business Transaction Match Rule - ASP.NET

Name: My ASP.NET Transaction

Enabled:

Priority: 1

Transaction Match Criteria | Split Transactions Using Request Data

Method: GET

URI: Contains Inventory

Split the transaction on the category parameter.

**Transaction Match Criteria** | **Split Transactions Using Request Data**

Split Transactions using request data ?

Use the first [ ] segments in Transaction names

Use the last [ ] segments in Transaction names

Use URI segment(s) in Transaction names  
Segment Numbers [ ] *Enter a comma separated list of parameter numbers (e.g. 1,3,4)*

Use a parameter value in Transaction names  
Parameter Name [ category ]

4. Click **Save**.

After the agent receives the updated configuration, it discovers the new business transaction and displays it in the [Business Transactions List](#).

Name	Health	Response Time (ms)	Calls
My ASP.NET Transaction.electronics		12	37

## Learn More

[Configure Business Transaction Detection for .NET](#)

[Web Entry Points](#)

[Configure Business Transaction Detection](#)