

Developing an add-on AppTier

This section includes the following topics:

- [About developing an Add-on AppTier](#)
- [About designing an Add-on AppTier](#)
- [About functional design of an Add-on AppTier](#)
- [About technical design of an Add-on AppTier](#)
- [About developing and testing the Add-on AppTier installation package](#)
- [About developing and testing the Insight SDK Collector](#)
- [About testing the complete Add-on AppTier solution](#)
- [About packaging, deployment, and distribution of an Add-on AppTier](#)

About developing an Add-on AppTier

To describe the Add-on AppTier development process, we will use an imaginary example of an Add-on AppTier application named Generic Mail Server (GMS). GMS provides email services to organizations and enterprises. It was developed by an independent software vendor here-named GMS-ISV.

GMS users are recognized by the application through their login user name. GMS users can log into the application using various devices and from various locations, such as: from their workstation, home desktop computer, laptop computer, or even their cell phone. The system monitors the user's current connection by logging the IP network address for the client device. GMS users can send and receive mail messages which can include attachments such as: documents, pictures, and various other types of files.

There are three options available for developing an Add-on AppTier:

- **Third-party vendor.** Develops and/or publishes a software application which enables its customers to monitor and tune performance for a locally installed application through the Precise environment.
For example, Independent Software Vendor (GMS-ISV) which owns and publishes the Generic Mail Server application, may decide to develop a GMS Add-on AppTier.
GMS application owners using Precise products can plug in the GMS Add-on AppTier to their local Precise installation to enable performance monitoring of their GMS application.
The GMS-ISV Research and Development team must develop the GMS Add-on AppTier and make the necessary changes to their GMS application so that performance data can be received and displayed through Precise products, such as: StartPoint, AdminPoint, Insight, and Report Manager.
- **Precise customer.** Develops an Add-on AppTier to monitor local installations or home-grown applications.
For example, GMS-ISV may decide not to develop an Add-on AppTier for GMS. GMS users that also use Precise products can still develop a customized GMS Add-on AppTier by developing tools and utilities that externally monitor the activity of their local GMS application, which will provide the performance measurements to be displayed in the Precise environment.
For this situation, the Precise customer's own software development team needs to research how to externally monitor the GMS application. This team will also develop the GMS Add-on AppTier, including the necessary tools and utilities that externally monitor local GMS application activities and provide the performance measurements to be displayed in the Precise environment.
- **Precise professional services.** Precise professional services can tailor solutions for Precise customers by developing Add-on AppTiers for various external applications Precise customers use.

About the development process

Developing an Add-on AppTier is a complex process that requires completing the following software development phases: design, code development, testing, packaging and distribution.

The following is a list of phases you must perform for successful development and delivery of an Add-on AppTier.

- **Design the Add-on AppTier.** The Design phase studies both functional and technical aspects for all components of the Add-on AppTier. Prepare a specifications document with all of the details.
The Add-on AppTier developer must determine the following regarding the data:
 - What information will be collected from the monitored application
 - How will the information be extracted or sampled
 - How will the information be stored and displayed in the Precise product suite
- **Develop and test the Add-on AppTier installation package.** The Add-on AppTier installation package consists of a set of XML files that contain metadata. The XML files define the Add-on AppTier to the Precise product suite and let Precise products correctly parse and display data they receive from an Add-on AppTier Collector.
Testing the installation package lets the Add-on AppTier developer examine how Add-on AppTier information will be displayed to the end user through Insight and Report Manager products. After testing, the developer can redesign the Add-on AppTier to tweak and optimize displays.
- **Develop and test the SDK Collector.** The Add-on AppTier installation package and the detailed specifications document (for expected deliverables from the Add-on AppTier Collector) should be ready almost in parallel. Next the Add-on AppTier developer must write code which complies with Precise standards for data format.
The Add-on AppTier developer has two options for developing code:
 - Write code for an Add-on AppTier Collector, or
 - Write code for an ARM API (through an Insight SDK dynamic extension library)
- **Test the completed Add-on AppTier solution.** The Add-on AppTier or Insight SDK Collector should now be functional and the Add-on AppTier installation package complete. The Add-on AppTier installation package specifies what data is expected from the Add-on AppTier or Insight SDK Collector.
Perform integration testing of the complete Add-on AppTier solution to verify correct data flow and communication.
Consider other aspects, such as: usability, performance, and scalability.
- **Package, deploy, and distribute the Add-on AppTier.** After testing is complete for the entire Add-on AppTier solution, plan how the installation package and Add-on AppTier or Insight SDK Collector should be packaged and distributed to end users.
Consider other aspects, such as: documentation, versioning, licensing, distribution, and support.

About designing an Add-on AppTier

There are two stages in Add-on AppTier design: functional and technical design.

During functional design, the Add-on AppTier developer needs to answer the following questions:

- Which information should be displayed in the Precise environment?
- How will the information be presented to the end user in Insight and Report Manager?

During technical design, the Add-on AppTier developer needs to answer the following questions:

- How will the data be sampled from the monitored application?
- How will the data be stored in the PMDB?

About prerequisites for Add-on AppTier design

To design an Add-on AppTier well, the Add-on AppTier developer must be knowledgeable in the following areas:

- **Monitored application.** Know how the monitored application operates and what is the best method for measuring performance for the application. Be able to answer the following questions regarding the monitored application:
 - What are common performance problems?
 - What are common complaints from users?
 - What types of performance measurements can be collected?
 - Does the monitored application support the ARM standard or does it support other standards and APIs?
 - Are there any external tools that can monitor the monitored application's performance?
- **Precise processing and display.** Understand how Insight and Report Manager display performance data from various technologies. Be able to answer the following questions:
 - What are the strengths in Insight and Report Manager and how best to utilize them?
 - What are the weaknesses in Insight and Report Manager?

About typical Precise AppTier components

A typical Precise AppTier contains several types of entities and counters.

- **Entities.** Entities identify the data giving it transactional context. There are three kinds of entities:
 - In client/server applications, entities can identify the client- or server-side machine, such as: login user name, IP address for client or server machine, and others.
 - Entities also identify information that is passed between the client and server. The passed information can be a message name, email title, transaction name, and others.
 - In self-contained applications (not client-server applications), these entities describe a basic (atomic) recurring operation or function that can be used to categorize application performance, such as: method or function name.
 - These entities are activity definition entities. Multiple entities can together define a single activity. However, there is always one entity that is classified as "the activity."
 - Entities can also identify global settings that are not tied to a specific transaction and effect the monitored application's operation. These entities can help identify problematic settings. For example, the same transaction can operate differently when activated online or in batch mode.
- **Counters.** Counters represent the performance measurement of activities. There are four kinds of counters:
 - **Service Time counters.** These counters measure the monitored application's working time to complete an activity. In its most basic form, there is one "work time" counter that measures the entire application's performance from the moment the activity begins until its completion. Service Time can sometimes be broken down to measure several separate functions that operate on the activity. For example, if the application is multi-tier, there can be both application and database times. Or perhaps the operation is passed to an external resource, such as: authentication or security checks. Breaking down the Service Time helps you discover which players are concerned with servicing a specific activity, and helps identify which player may be responsible for an existing bottleneck.



Service Time defines the working time of the monitored application itself. It does not include network times and fat client times.

- **Network time.** Network time measures the time it takes for the activity to pass through the network between the client and server. Network time can be calculated by the Add-on AppTier Collector, or by integrating the Add-on AppTier with the Insight Network Collector.
- **Statistical counters.** These counters are measurements that do not describe the working time of an activity. For example a message can be divided into several packets, or an email can be of a specific length and can contain one or more attachments. Statistical counters are often used to characterize poor performing activities. For example, the Precise user may find that email messages with large attachments are causing high network times.
- **Executions.** An Executions counter specifies the number of invocations for the same activity that occurred in one time slice (every 15 minutes). There are different Executions counters for every mutation of entity values. For example: if the same email message is sent by two different users, there will be two records in the XML file from the Add-on AppTier Collector, each file having a different Executions counter.

About entity classifications

Entities and their attributes identify information that is passed between the client and server. There are two aims for classifying entities:

- **Identify entities for pre-load processing.** Insight performs special processing on specific classified entities, before the data is loaded into the PMDB. The pre-load process includes the following calculations:
 - **Calculate Service Level Agreement (SLA) breaches for activities.** For every activity invoked, Insight calculates the average service time in the time slice as total reported service time divided by the number of executions. Insight compares the average service time to SLA definitions and assigns a color code (green, yellow, or red) to all activity executions in the time slice. For this aim, Insight needs to identify the entity that is classified as the activity.
 - **Calculate the location of the Client IP address.** If an entity with a Client IP classification exists, Insight will look for and assign a location definition to that entity.
- **Cross-AppTier comparison.** Classifying entities enables Insight to display entities from different AppTiers side-by-side for comparison. For example, if an entity is classified as a Client IP, Insight will display this entity side-by-side with the Client IP entity of an Precise for Web product. Insight displays a side-by-side comparison of different AppTiers in the Cross-AppTiers workspace.

The following is a complete list of entity classifications as they appear in the Insight Display list:

- Activities
- Super Activities
- Client IPs
- Servers
- Instances
- Users
- Locations

About defining an Activity or Super Activity

An Activity is the atomic (most basic) unit (entity) which defines an action. The following are some examples of an activity:

- SQL statements in a database tier
- URL address in a Web tier
- Method in a Microsoft .NET or J2EE application tier
- Transaction for SAP in an application tier

A Super Activity is an entity which invokes other Activities. For example, a program that activates SQL statements in a database tier is a Super Activity.

About counter classifications

Counters are classified into Service Time, Statistical counters, and Executions, all used for similar purposes:

- **Identify counters for pre-load processing.** To calculate an SLA branch, Insight must identify Service Time counters as well as the Executions counter.
- **Cross-AppTier and Over Time comparison.** Insight and Report Manager use Service Time counters to compare the performance of different AppTiers side-by-side.

About Grouping

Grouping is a feature that enables Precise users to aggregate multiple values of performance measurement for one entity into a single line in a graph, according to a textual pattern. You use this feature to group entity values according to business functions.

After you have defined a grouping for an entity, a new graph and table appear in Insight. For example, Insight displays performance calculations for the Senders grouping. Each displayed line indicates the sum of all performance calculations for Senders that belong to this group.



Report Manager lets you define custom reports for grouped entities. Groups for Client IP entities are called Locations and are defined using a separate settings dialog in AdminPoint.

About Insight and Report Manager standards and restrictions

Insight and Report Manager impose certain standards and restrictions to AppTiers:

- **Flat hierarchy of entities.** Insight and Report Manager do not recognize a hierarchy of entities. There is no tree view and all entities are displayed in identical graphs and tables.
- **All counters are applicable to all entities.** Insight and Report Manager compare all entities according to the same counters; this design is best suited for comparing entities according to performance measurements. This design is less suited for displaying counters that are not applicable to specific entities, such as: instance statistics.
- **Maximum of 10 entities and 10 counters.** Based on the two previous standards for entities, Insight and Report Manager always display all entities compared to all counters. If there are many entities and/or many counters, the screen display or report can become crowded and cumbersome. As a general guideline to follow, create no more than 10 entities and 10 counters.

About functional design of an Add-on AppTier

The functional design of an Add-on AppTier involves setting how the Add-on AppTier will be displayed in Insight and Report Manager; whether to set it as a standalone AppTier, or in multi-AppTier environments.

Consider the following issues when performing a functional design:

- **Familiarize yourself with Insight and Report Manager.** View standard Precise technologies through Insight screens and Report Manager reports.
See how the products display performance measurements in the various graphs and tables and get a general idea of how you want the monitored application to display.
- **Decide on the entities and counters for the monitored application.** If the monitored application is typically part of multi-tier environments, you will want to classify your entities and counters so that Precise users can benefit from the Insight Cross-AppTier workspace display and the Report Manager Environment Profile report.

About guidelines for the functional design of an Add-on AppTier

The following are guidelines you should follow for entities:

- **Mandatory entities: Instance, Server and Activity.** Instance and Server entities are mandatory because they are used in AdminPoint when you define a new instance for the Add-on AppTier.
An entity with an Activity classification must also be defined. The Activity entity is the basis for SLA calculations and other processing.
- **Functional attributes.** There are four functional attributes for an entity to consider:
 - **Entity name.** The Entity name is used as the title for Insight and Report Manager graphs. You need to specify the entity name in singular and plural variations.
 - **Entity classification.** The Entity classification is useful for multi-tier environments to compare this entity against similar entities in different AppTiers.
 - **Is it mandatory?** In other words, does the entity contain a value for every activity, or maybe the entity is not applicable in certain cases? Instance, Server, and Activity are mandatory entities; every record in the Add-on AppTier Collector file will have a value for these entities.
 - **Do you want to allow grouping for the entity?** Grouping is a sensible solution if the entity's collected values can be categorized by Precise users according to a business or technical function. It allows Precise users to group mail Senders according to departmental categories.
However, the Generic Mail Server Add-on AppTier also has an entity called Message Type that defines regular emails, meeting schedules, and others. The Message Type entity does not support the grouping feature.

The following are guidelines you should follow for counters:

- **Mandatory counters: minimum of one Service Time counter.** There must be at least one counter defined with the Service Time classification. The Service Time counter is the basis for all SLA calculations, Cross-AppTier comparisons, and other processing.
- **Optional: Executions.** The Executions counter is not mandatory.
If not specified, Insight treats every record in the Add-on AppTier Collector file as one execution.
- **Functional attributes.** There are four functional attributes for a counter to consider:
 - **Counter name.** The Counter name is used as an Insight Tool Tip in the Compared by drop-down list and in the report Manager Y-axis legend.
 - **Counter classification.** The Counter classification defines Service Time counters, the Executions counter, and Statistical counters.
 - **Counter color.** Both Insight and Report Manager support a pre-defined color palette.
 - **Unit of measure.** This attribute affects the display format for counter values in Insight and Report Manager. Service Time counters are always time based while the Executions counter is always a number.
However, Statistical counters can represent either a number of Bytes. If Bytes, they are summed as KiloBytes, MegaBytes, and so on.

The following are general guidelines:

- **Add-on AppTier name.** The Add-on AppTier name is used during plug-in installation of the Add-on AppTier and in StartPoint, AdminPoint, Insight, and Report Manager screens.
- **SLA applicability.** You can give Precise users the ability to define SLA levels for activities.
If provided, Precise users can define SLA levels in AdminPoint for the average Service Time for activities. Insight will calculate the SLA breaches for performance data that is received from the Add-on or Insight SDK Collector.

About technical design of an Add-on AppTier

The technical design of an Add-on AppTier involves setting how data will flow from the Add-on AppTier or Insight SDK Collector to Insight, and how the data will be stored in the PMDB.

Consider the following issues when performing a technical design:

- **Use ARM of file-based collection.** You must decide what mode of collection to use. See [About guidelines for the technical design of an Add-on AppTier](#).
- **Design the Collector file.** Whether you use ARM (Application Response Measurement) or file-based collection, the Collector XML file becomes the vehicle by which the Add-on AppTier or Insight SDK Collector provides data for Precise installation.
Insight must know how to correctly read and parse the Collector file before it is loaded into the PMDB.

About guidelines for the technical design of an Add-on AppTier

The following are guidelines you should follow for the technical design:

- **ARM collection mode.** ARM (Application Response Measurement) is an industry standard that is growing in popularity.
Implementing the ARM standard in the monitored application will open up the monitored application to all monitoring and tuning tools that support this standard. It also allows the tools to probe the monitored application and display performance calculations and perform application analysis.
Using the ARM standard API solution involves code instrumentation. This is possible only if the Add-on AppTier developer has access to the source code for the monitored application. For example, the Generic Mail Server vendor can implement the ARM standard API by incrementing

the application's code.

Using the ARM standard API, the Insight SDK Collector creates a Collector XML file (a complex process) and aggregates the information after every time slice (every 15 minutes).

Currently you can implement the ARM standard API for an Add-on AppTier using C language only.

For more details regarding the ARM standard, refer to <http://www.opengroup.org/management/arm/>.

- **File collection mode.** In the file collection mode, the task of developing the Add-on AppTier Collector rests on the shoulders of the Add-on AppTier developer.

The Add-on AppTier developer can monitor the application and collect performance data any way that suits his needs. He is not limited by restrictions, such as: API standards, language support, platform support, and so on.

The Add-on AppTier Collector must create Collector XML files using Precise format and file naming conventions.

See [About ETD file application-level entities](#) for more information.

The Add-on AppTier Collector aggregates performance measurement into 15-minute time slices that are then written into separate Collector files.

- **Entity and counter technical attributes.** There are two technical attributes to consider:
 - **Database column name.** This is the name of the database column in the PMDB that stores entity or counter values.
 - **XML tag name in the Collector file.** This is the name of the XML tag in the Add-on AppTier or Insight SDK Collector file that holds values for the entity or counter.

About developing and testing the Add-on AppTier installation package

The Add-on AppTier plug-in installation package is a series of metadata XML files that contain all functional and technical attributes for Add-on AppTier entities and counters.

Testing the Add-on AppTier installation package gives the Add-on AppTier developer his first glimpse into how the Add-on AppTier will display in Insight and Report Manager. The Add-on AppTier installation package also contains the technical specifications to the Add-on AppTier or Insight SDK Collector file.

About prerequisites for developing an Add-on AppTier installation package

You need the following prerequisites to develop and test the Add-on AppTier installation package:

- Functional and technical specifications for the Add-on AppTier
- Add-on AppTier installation package specifications, as described in [Add-on AppTier installation package descriptor file](#).
- Insight and Report Manager FocalPoints installed

About testing the Add-on AppTier installation package

Perform the following to test the Add-on AppTier installation package:

- **Plug-in the Add-on AppTier package.** Plug-in the Add-on AppTier installation package using AdminPoint. This will test whether the installation package is correct and complete. See [About installing the Add-on AppTier plug-in installation package](#) for more information.
- **Manually create a Collector file and load it to the PMDB.** After plugging-in the Add-on AppTier installation package has finished, you need to wait until data is loaded into the PMDB before any information displays through Insight or Report Manager. Consequently for testing, you must supply an Add-on AppTier Collector XML file with a sample file. In most cases, a Collector XML file will not exist until you develop the Add-on AppTier Collector. Manually create the Collector XML file according to the XML tag names specified in the Add-on AppTier installation package, and according to file name conventions. See [About the XML Collector file](#) for file name conventions. After your Collector XML file is successfully loaded into the PMDB, and Insight and Report Manager display the correct information, the sample Collector file can be used as a reference for the following phases in developing the Add-on AppTier or Insight SDK Collector.

About developing and testing the Insight SDK Collector

After testing the Add-on AppTier installation package and a sample Collector file is available as reference, you can write the code for an Insight SDK Collector.

The Insight SDK Collector collects application-specific information using an ARM (Application Response Measurement) standard API (Application Programming Interface) in the application. It correlates the transactions it receives from the Insight ARM extension dynamic library, within the application, and creates performance files.

Initializing the Insight ARM extension dynamic library

Initializing the Insight ARM extension dynamic library begins when the Add-on AppTier application calls to the `arm_init` function.



Before installing the Insight ARM extension dynamic library, copy the original ARM dynamic library to another directory.

To initialize the Insight ARM extension dynamic library

1. Create or attach to the shared memory segment holding the configuration.
2. Connect to the IPC channel. The IPC channel passes transactions information from the Insight ARM extension dynamic library to the Insight SDK Collector. The IPC channel (UDP port) number is stored in the shared memory.
3. Initialize the application and user-defined properties within the parameters for the `arm_init` function.
4. Initialize the transaction hash. This hash includes all user-defined transactions defined while the Add-on AppTier application is running.

About testing the complete Add-on AppTier solution

Testing the complete Add-on AppTier solution involves completing the following cycle:

- Plugging-in the Add-on AppTier using the Add-on AppTier installation package
- Defining instances and invoking the Insight SDK Collector

Place Collector files in the correct directory, where the Precise harvester automatically puts the files into the appropriate directory on the Precise FocalPoint Server.

Other aspects, such as: performance and scalability for the Add-on AppTier can now be addressed.

About packaging, deployment, and distribution of an Add-on AppTier

These are the final steps in developing the Add-on AppTier solution.

Other aspects, such as: documentation, versioning, licensing, distribution and support should now be addressed.

About Add-on AppTier registration

Before distributing an Add-on AppTier installation package to Precise users, send the newly developed and tested Add-on AppTier installation package to the Precise research and development team for review, validation, and certification.

Your new Add-on AppTier will be allocated a unique tech-code and a user interface titles range to be specified in the ETD file.



For development and testing, you can use a list of pre-allocated tech-code and a user interface titles range specified in, see [About the titles file or files](#).

The following information must be sent to the research and development team:

- A high-level document describing the monitored application
- A high-level description describing the purpose and functionality of the Add-on AppTier solution
- If using the ARM API option, describe the instrumented transactions and their properties (as passed to the ARM API)
- The Add-on AppTier installation package
- Optional: target customer and use cases
- Contact details for the Add-on AppTier developer

Email the required information to: dl-rnddevelopment-tpmteamleaders@precise.com